NASA-CR-178001

NASA Contractor Report 178001

NASA-CR-178001
19860004487

MINIMAL NORM CONSTRAINED INTERPOLATION

Larry Dean Irvine

OLD DOMINION UNIVERSITY
Norfolk, Virginia

Grant NAG1-363
September 1985

NASA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

TABLE OF CONTENTS

This Page Intentionally Left Blank

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.

This Page Intentionally Left Blank

ABSTRACT

MINIMAL NORM CONSTRAINED INTERPOLATION

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

In computational 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.

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 $0 \leqq y_i \leqq 1$ 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

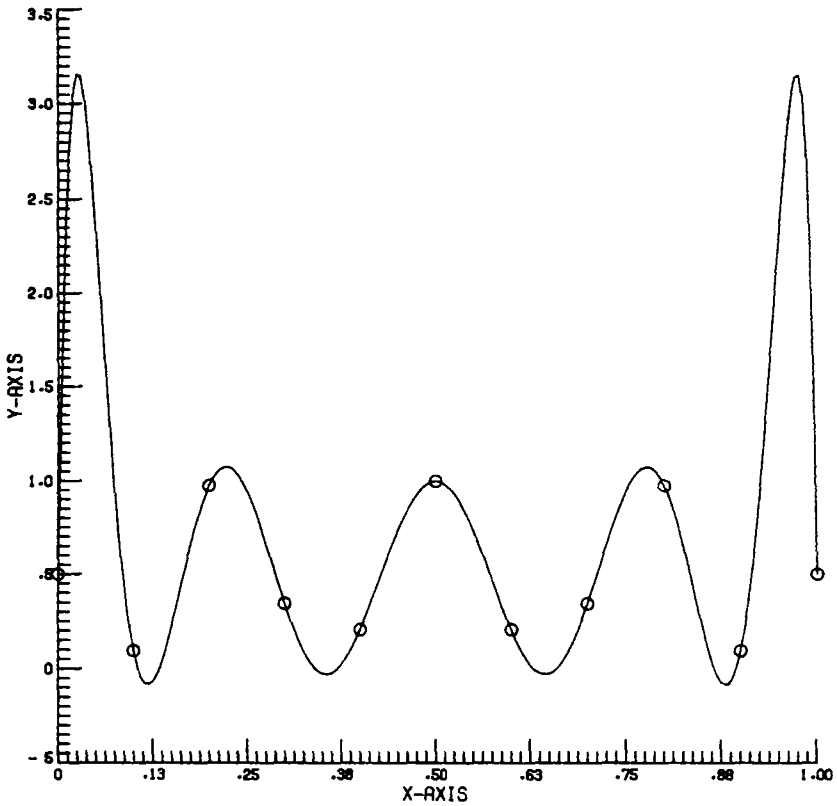

Figure (1.1): The Polynomial Interpolant.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 $\tau_i = t_i$)

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

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

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

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

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

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

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

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).

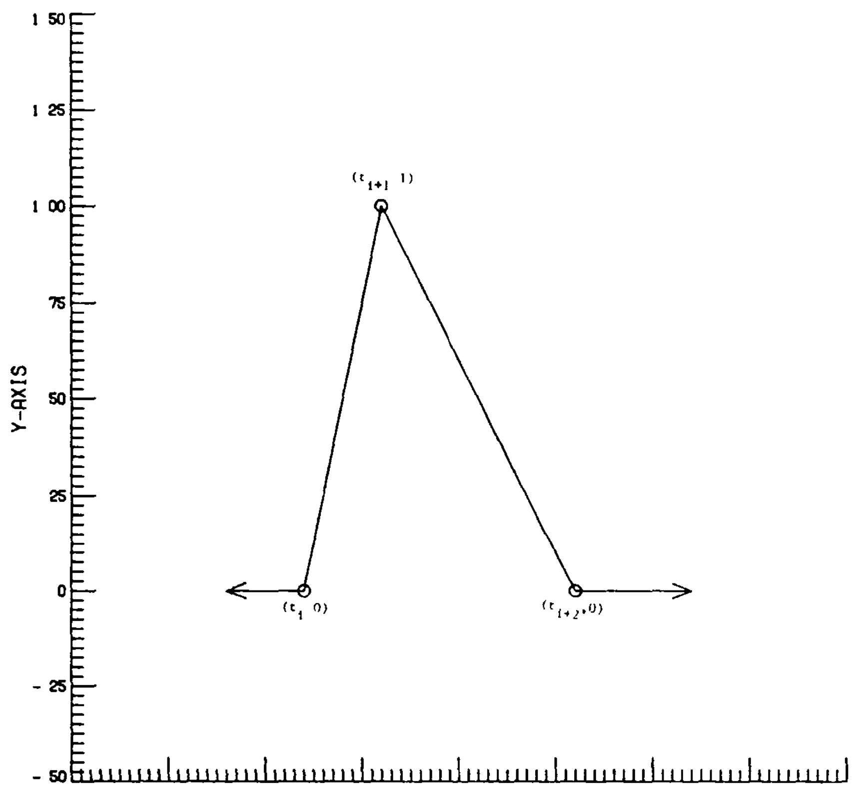

Figure (1.2): The Normalized Linear B-spline.

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

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

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

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

and characterized by

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

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 $S: = \text{span}(N_1, N_2, \ldots, N_m)$ 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) an element s_o in S such that

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

with $y - s_o$ in the orthogonal complement of S. This implies $T(y - s_o) = \theta$ or $(y - s_o) \varepsilon W$. 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

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

(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 $\underline{x} \varepsilon R^m$ we have

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

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

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

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

$$\geqq 0$$

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 $i=1,2,\ldots,m$

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

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

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

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

We also compute for $i=1,2,\ldots,m-1$

$$A_{i,i+1} = A_{i+1,i} = \int_{t_{i+1}}^{t_{i+2}} N_i N_{i+1}(t)dt = (t_{i+2} - t_{i+1})/6.$$

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 $i=1,2,\ldots,n-1$ and $j=0,1,2,3$. Then

$$f_*(t) = \frac{p_{*_i}{}^{(2)}(t_i)}{0!} + \frac{p_{*_i}(t_i)}{1!}(t-t_i) + \frac{p_{*_i}{}^{(2)}(t_i)(t-t_i)^2}{2!} + \frac{p_{*_i}{}^{(3)}(t_i)(t-t_i)^3}{3!} \qquad (1.14)$$

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

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

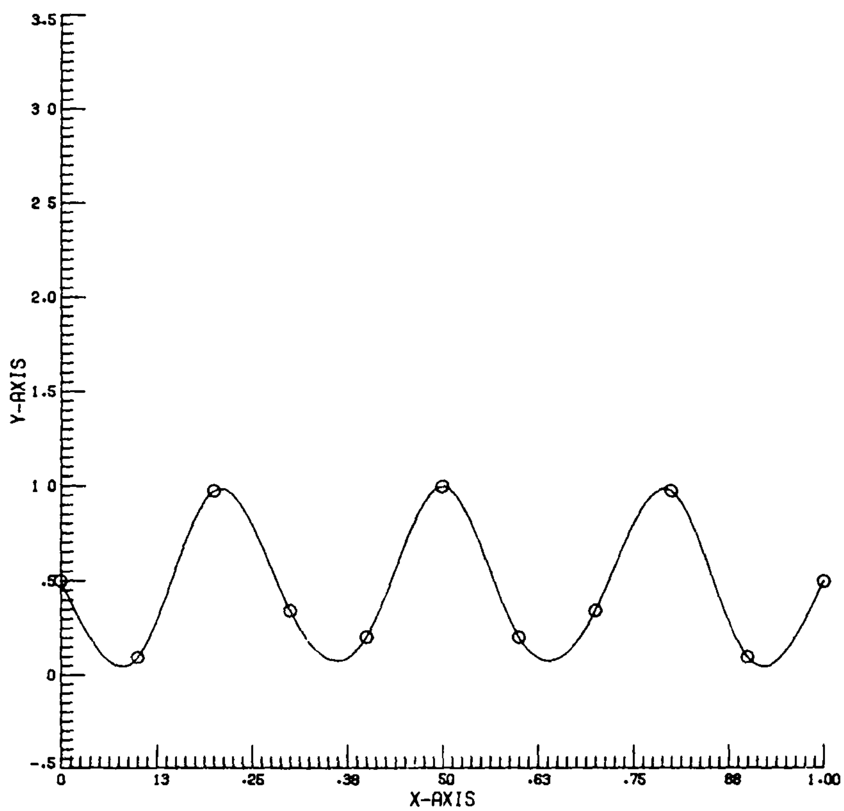

Figure (1.3): The Natural Cubic Spline Interpolant.

consider the problem

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

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

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

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

or

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

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

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

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.

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

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

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

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

Find $g_* \in G(p)$ such that $\|g_*\|_p \leqq \|g\|_p$ for all $g \in G(p)$. (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

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

via

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

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,

$$\|\lambda\|_S \leq \|g\|_p \text{ for all } g\varepsilon G(p). \tag{2.4}$$

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 $\hat{g}\,\varepsilon\,G(p)$ such that

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

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

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

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

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

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

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

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

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

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

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

and

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

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

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

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 $1 \leq i_1 < i_2 < i_3 \leq n$. Equivalently,

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

(where f is any interpolant to the data) or

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

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.

(There are convex data for which A is empty. For example, let $y_1 = |t_1|$ 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 $f \varepsilon A$ then $T(f^{(2)}) = \underline{d}$ where $T:L_2[a,b] \to R^m$ is given by $(Tg_1) := (g,N_1)$. Hence if

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

then problems

$$\text{Find } f_* \varepsilon A \text{ such that } ||f_*^{(2)}||_2 \leqq ||f^{(2)}||_2 \text{ for all } f \varepsilon A \quad (3.1)$$

and

$$\text{Find } g_* \varepsilon B \text{ such that } ||g_*||_2 \leqq ||g||_2 \text{ for all } g \varepsilon B \quad (3.2)$$

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_ $k_o \varepsilon K$ _such that_

$$||x - k_o|| \leqq ||x - k|| \quad \text{for all} \quad k \varepsilon K$$

Furthermore, k_o _is characterized by_

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

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

$$(g_*, g-g_*) \geqq 0 \quad \text{for all} \quad g \varepsilon B. \quad (3.3)$$

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

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

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

<u>Proof</u>: 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

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

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].

We begin by considering the problem

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

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

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

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

At a minimum of L we must have

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

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

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

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

or

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

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

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

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

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

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

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

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

$$= 1$$

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

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

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

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

We conclude that $\sum_{j=1}^{m} \alpha_j N_j$ is nonpositive, but not identically zero. Since we have assumed there exists $\hat{g} \varepsilon B$ 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 $\hat{g} \varepsilon B$, 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 $g \varepsilon B$ 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

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

which implies that $\alpha_2 = -\infty$. This is equivalent to the solution being identically zero on [1,3]. In fact, any $g \varepsilon B$ 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

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

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

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

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

Two integrations gives us

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

and

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

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

$$\tau_i = t_i - \beta_i \Delta t_i / \Delta\beta_i \tag{3.14}$$

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

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

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

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

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

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

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

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

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

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

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

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

$$
\begin{aligned}
p_{*i}(t_i) &= y_i \\
p_{*i}^{(1)}(t_i) &= \Delta y_i/\Delta t_i \\
p_{*i}^{(2)}(t_i) &= 0 \\
p_{*i}^{(3)}(t_i) &= 0.
\end{aligned}
\tag{3.18}
$$

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

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

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

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

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.

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

$(\sum_{k=1}^{m} \alpha_k N_k)_+$ 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

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

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

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

$$\geqq 0$$

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

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

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

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

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

where S(k) is the support of $\left(\sum_{j=1}^{m} x_j^{(k)} N_j\right)_+$. 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 $\left(\sum_{j=1}^{m} x_j^{(k+1)} N_j\right)_+$ 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 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 convenience (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

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

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.

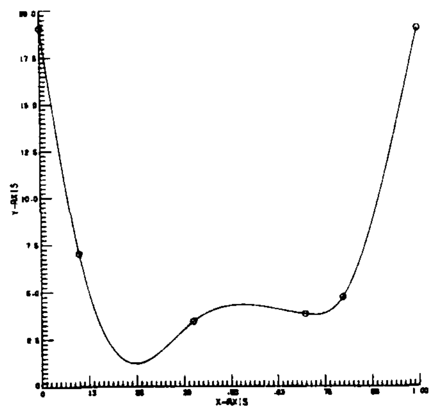

Figure (3.1): The Natural Cubic Spline Interpolant.

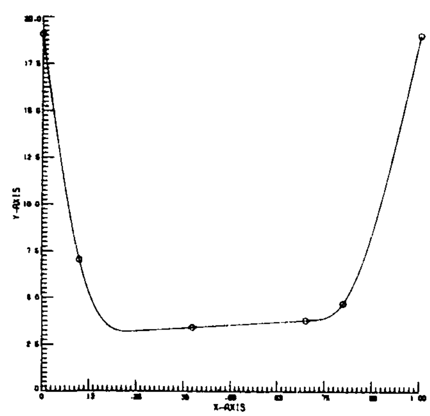

Figure (3.2): The Convex Cubic Spline Interpolant

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	$.39 \times 10^{-11}$	-----	-----

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

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

we conclude that the problems

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

(the problem of finding the smoothest concave interpolant) and

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

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

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

then $g_* = -\left(\sum_{j=1}^m \alpha_j N_j\right)_-$. 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

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

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

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

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

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

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

We conclude that the problems

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

and

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

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 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

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

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

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

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

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

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 $i=1,2,\ldots,m$.

The following theorem is modeled after theorem (3.2).

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

Note that if $\underline{x}^{(0)}$ is given by $x_i^{(0)}$ = 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

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

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

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

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 $\beta_i \leqq 0$ and $\beta_{i+1} \leqq 0$, 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 $\beta_i \geqq 0$ and $\beta_{i+1} \geqq 0$.

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.

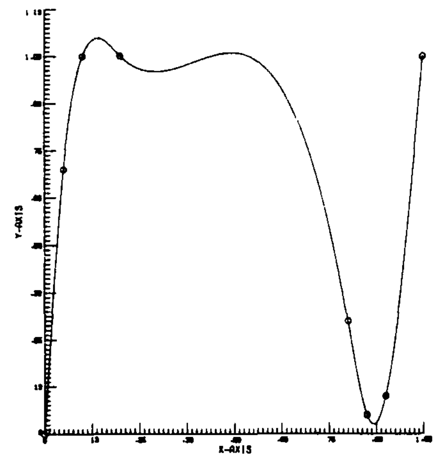

Figure (4.1): The Natural Cubic Spline Interpolant.

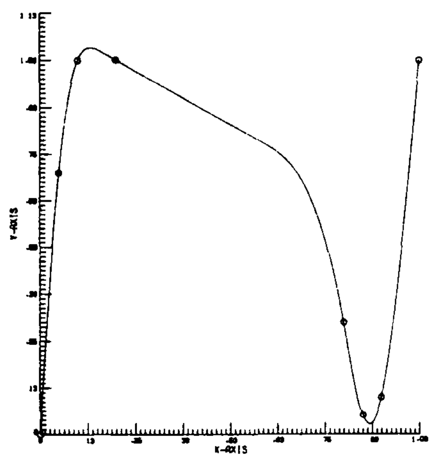

Figure (4.2): The Shape-Preserving Cubic Spline Interpolant.

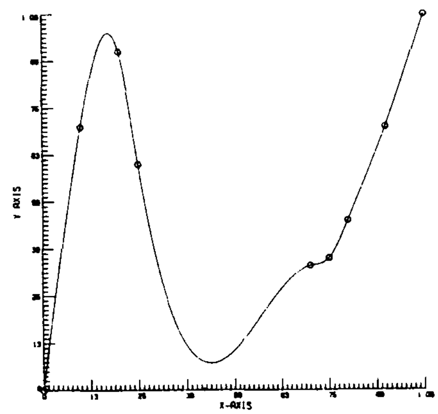

Fıgure:(4.3): The Natural Cubıc Splıne Interpolant.

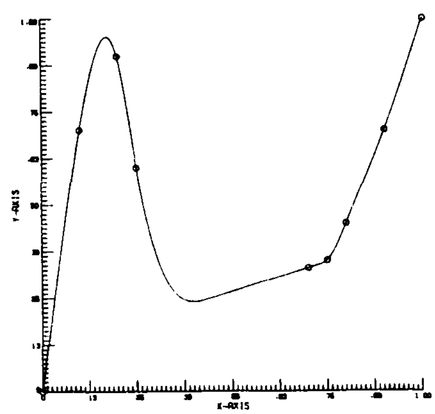

Figure (4.4): The Shape-Preservıng Cubic Splıne Interpolant.

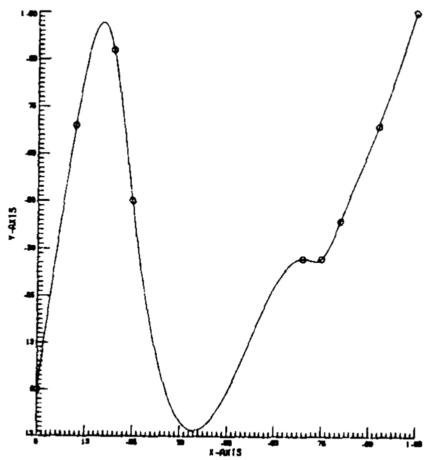

Figure (4.5): The Natural Cubic Spline Interpolant.

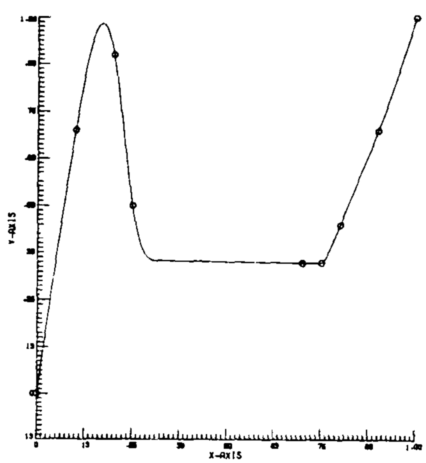

Figure (4.6): The Shape-Preserving Cubic Spline Interpolant.

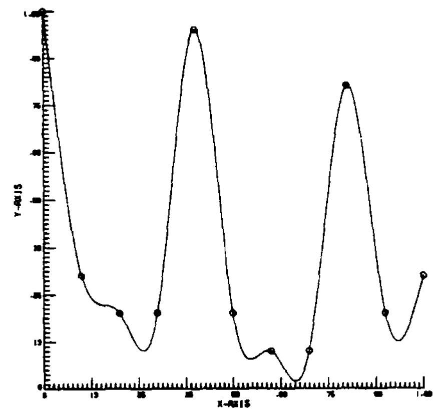

Figure (4.7): The Natural Cubic Spline Interpolant.

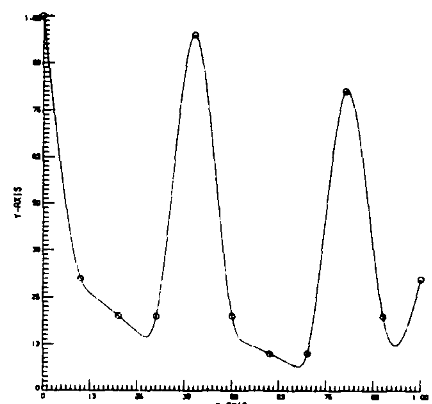

Figure (4.8): The Shape-Preserving Cubic Spline Interpolant.

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}

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}

Table 4.3

Iteration Number	$\lVert F(\underline{x}^{(n)}) - \underline{d} \rVert_2$
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}

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}

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

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

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

$$Tx_n = \underline{d} \tag{5.2}$$

and

$$\|x_n\| \leq \gamma + 1/n \tag{5.3}$$

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

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

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

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

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

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

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

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

For each n there exists $x_n \varepsilon C$ 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 $x \varepsilon C$. Hence $\|x\| \leq 1$ and $Tx = \underline{z}$. This completes the proof of the proposition.

We define

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

Equivalently,

$$\begin{aligned}\gamma^* &= \inf\{\gamma : \text{There exists } x \varepsilon C \text{ such that} \\ &\qquad Tx = \underline{d} \text{ and } \|x\| \leq \gamma\} \\ &= \inf\{\|x\| : x \varepsilon C \text{ and } Tx = \underline{d}\} . \end{aligned} \tag{5.5}$$

By lemma (5.1) we know that there exists $x_* \varepsilon C$ 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

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

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 $y \varepsilon Y$. 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 $\{x \varepsilon C : \|x\| \leq 1\}$.

Since C is weak* closed, the supremum is attained at some element of $\{x \varepsilon C : \|x\| \leq 1\}$; that is, for any $y \varepsilon Y$ there exists an x (a function of y) such that $x \varepsilon C$, $\|x\| \leq 1$, and $\rho(y) = x(y)$. In fact we have $\|x\| = 1$ unless $x = 0$. 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 $\{x \varepsilon C : \|x\| \leq 1\}$ 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

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

and hence

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

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

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

By definition

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

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

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

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

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

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

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

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

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

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

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

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

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 $x_m \in C$

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 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 $\underline{y} \varepsilon G(\gamma^* - \varepsilon)$ 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

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

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

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

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

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

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

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

This completes the proof of the lemma.

Let λ be a linear functional defined on the subspace

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

by

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

so that (5.6) can now be written

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

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

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

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

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

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

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

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

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

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

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

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

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

<u>Proof</u>: Let $z = z_o + tu$ where $t > 0$ and $u \varepsilon Z$. 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 $u \varepsilon Z$, 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 $u \varepsilon Z$. Taken together, (5.16) and (5.17) imply $\lambda = (\nabla f)(z_o)$.

<u>Corollary (5.11)</u>: <u>If</u> ρ <u>is differentiable at</u> $y_o \varepsilon Y$, <u>then</u> $\rho'(y_o) = x_o$.

<u>Proof</u>: Since $\rho(y_o) = x_o(y_o)$ and $x_o(y) \leqq \rho(y)$ for all $y \varepsilon Y$, we have

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

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

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

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

and hence

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

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 the next theorem reveals.

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

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

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

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

<u>is an interpolant of minimal norm</u>.

<u>Proof</u>: 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

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

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

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

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

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

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

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

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

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

Equivalently, we have

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

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

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

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

<u>Proposition (5.13)</u>: <u>If d is in the relative interior of</u>

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

<u>then there exists a vector β such that</u>

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

<u>Proof</u>: 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

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

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

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

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.

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.

Appendix A

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

```
      PROGRAM UNCON(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C       WE FORM THE NATURAL CUBIC SPLINE INTERPOLANT.
C
      INTEGER N,M,I
      REAL T(50),F(50),D(50),X(50),A(50),PP(4,50)
      REAL AA(50),BB(50),CC(50)
C
C       THE ARRAYS (T) AND (F) - EACH OF SIZE M, THE NUMBER
C       OF DATA. POINTS - CONTAIN THE COMPONENTS OF THE DATA.
C       THE DATA FILE IS OF THE FOLLOWING FORM
C
C         M
C         T(1),F(I)
C         T(2),F(2)
C              .
C              .
C              .
C         T(M),F(M)
C
C       WHERE WE ASSUME (T) HAS STRICTLY INCREASING COMPONENTS.
C
      READ(3,*) M
      READ(3,*) (T(I),F(I), I=1,M)
      N= M-2
C
C       THE ARRAY (D) CONSISTS OF THE SCALED
C       SECOND DIVIDED DIFFERENCES.
C
      DO 100 I=1,N
      D(I)= ( F(I+2)-F(I+1) )/( T(I+2)-T(I+1) )
     C          - ( F(I+1)-F(I) )/( T(I+1)-T(I) )
  100 CONTINUE
C
C
C       THE SECOND DERIVATIVE OF THE NATURAL CUBIC SPLINE
C       INTERPOLANT IS A LINEAR COMBINATION OF LINEAR B-SPLINES.
C       WE CALCULATE THE COEFFICIENTS.
C
C
      AA(1)= 0.0
      BB(1)= (T(3)-T(1))/3.0
      CC(1)= (T(3)-T(2))/6.0
      DO 200 I=2,N-1
      AA(I)= (T(I+1)-T(I))/6.0
      BB(I)= (T(I+2)-T(I))/3.0
      CC(I)= (T(I+2)-T(I+1))/6.0
  200 CONTINUE
      AA(N)= (T(N+1)-T(N))/6.0
      BB(N)= (T(N+2)-T(N))/3.0
```

```
      CC(N)= 0.0
      CALL TRID(AA,BB,CC,D,N)
C
C
C
      A(1)= 0.0
      A(M)= 0.0
      DO 300 I=2,N+1
      A(I)= D(I-1)
300   CONTINUE
C
C
C       NOW WE COMPUTE THE NUMBERS PP(J,I) - THE VALUE
C       OF THE (J-1)ST DERIVATIVE OF THE NATURAL CUBIC
C       SPLINE INTERPOLANT EVALUATED AT T(I).
C
C
      DO 400 K=1,N+1
      DF= F(K+1)-F(K)
      DT= T(K+1)-T(K)
      DA= A(K+1)-A(K)
      PP(4,K)= DA/DT
      PP(3,K)= A(K)
      PP(2,K)= DF/DT - (A(K)/2. + DA/6.)*DT
      PP(1,K)= F(K)
400   CONTINUE
      PP(4,M)= 0.0
      PP(3,M)= 0.0
      PP(2,M)= 0.0
      PP(1,M)= F(M)
C
C
C
      DO 500 K=1,M
      WRITE(6,450) K,T(K),(PP(I,K), I=1,4)
450   FORMAT(5X,I5,5F14.6)
500   CONTINUE
C
C       WE CREATE A DATA FILE FOR PLOTTING THE (JDER)-TH
C       DERIVATIVE OF THE NATURAL CUBIC SPLINE INTERPOLANT
C       BY EVALUATING IT AT (MM) EQUALLY SPACED POINTS,
C       INCLUDING THE ENDPOINTS.  WE ASSUME THAT (JDER)
C       HAS VALUE 0, 1, 2, OR 3.
C
      JDER= 0
      MM= 201
      CALL DATAFL(T,PP,M,MM,JDER)
C
      STOP
      END
```

```
      SUBROUTINE DATAFL(TX,PP,LI,MM,JDER)
C
C       WE CREATE A DATA FILE FOR PLOTTING THE (JDER)-TH
C       DERIVATIVE OF THE PIECEWISE CUBIC POLYNOMIAL.  WE
C       ASSUME (JDER) HAS VALUE 0, 1, 2, OR 3.
C
      INTEGER LI,MM,JDER
      REAL TX(100),PP(4,100)
      LEFT= 1
      MMONE= MM - 1
      WRITE(4,*) MM
      XE= ( TX(LI)-TX(1) )/FLOAT(MMONE)
      DO 500 IP=1,MM
      XT= TX(1) + XE*FLOAT(IP-1)
C
C       WE FIND THE INTERVAL IN WHICH THE POINT (XT) LIES.
C
          IF ( LEFT .NE. LI ) THEN
            DO 200 IS=LEFT,LI-1
            IF ( XT .LT. TX(IS+1) ) GO TO 300
200         CONTINUE
300         CONTINUE
          END IF
          LEFT= IS
C
C       WE NOW COMPUTE THE VALUE OF THE POLYNOMIAL AT
C       THE POINT (XT) BY USING MESTED MULTIPLICATION.
C
      H= XT - TX(LEFT)
      FAC= 4.0 - FLOAT(JDER)
      YT= 0.0
      DO 400 M=4,JDER+1,-1
          YT= (YT/FAC)*H + PP(M,LEFT)
          FAC= FAC - 1.0
400   CONTINUE
      WRITE(4,450) XT,YT
450   FORMAT(F8.4,E18.9)
500   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRID(SUB,DIAG,SUP,B,N)
      INTEGER N,I
      REAL B(N),DIAG(N),SUB(N),SUP(N)
           IF (N.LE.1) THEN
           B(1)= B(1)/DIAG(1)
           RETURN
           END IF
      DO 111 I=2,N
           SUB(I)= SUB(I)/DIAG(I-1)
           DIAG(I)= DIAG(I) - SUB(I)*SUP(I-1)
           B(I)= B(I) - SUB(I)*B(I-1)
111   CONTINUE
           B(N)= B(N)/DIAG(N)
      DO 222 I=N-1,1,-1
           B(I)= (B(I)-SUP(I)*B(I+1))/DIAG(I)
222   CONTINUE
      RETURN
      END
```

Appendix B

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

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C       WE COMPUTE A SHAPE-PRESERVING INTERPOLANT
C       TO GIVEN DATA.
C
C
C       NOTE ON THE SIZE OF THE ARRAYS:
C       THE ARRAYS (T), (F), AND (A) MUST BE OF LENGTH
C       AT LEAST M, THE NUMBER OF DATA POINTS.  THE
C       ARRAY (TX) AND THE SECOND COMPONENT OF THE
C       ARRAY (PP) SHOULD BE OF LENGTH 2M.  THE ARRAYS
C       (X), (Y), AND (D) MUST BE OF LENGTH AT LEAST M-2.
C       THE ARRAY (ID) MUST BE OF LENGTH AT LEAST M-1.
C
C
      REAL T(50),F(50),X(50),Y(50),A(50)
      REAL TX(100),PP(4,100),TL,TR,EPS
      INTEGER M,N,ITMAX,I,J,IFLAG,MM
      COMMON D(50),ID(50)
C
C
C
C       THE ARRAYS (T) AND (F) - EACH OF SIZE M, THE NUMBER
C       OF DATA POINTS - CONTAIN THE COMPONENTS OF THE DATA.
C       THE DATA FILE IS OF THE FOLLOWING FORM
C
C           M
C           T(1),F(1)
C           T(2),F(2)
C               .
C               .
C               .
C           T(M),F(M)
C
C       WHERE WE ASSUME (T) HAS STRICTLY INCREASING COMPONENTS.
C
      READ(5,*) M
      READ(5,*) (T(I),F(I), I=1,M)
      N= M-2
C
C       (EPS) IS A SMALL POSITIVE NUMBER USED TO TEST FOR
C       CONVERGENCE IN NEWTON'S METHOD - SUBROUTINE (ZERO).
C       (ITMAX) IS THE MAXIMUM NUMBER OF ITERATIONS
C       WHICH WE PERMIT FOR NEWTONS METHOD TO CONVERGE.
C
      EPS= 1.0E-8
      ITMAX= 25
C
C       THE ARRAY (X) IS THE KNOT SEQUENCE (T) WITH THE
C       ENDPOINTS TL AND TR DELETED.
```

```
C
      TL= T(1)
      TR= T(M)
      DO 120 I=1,N
      X(I)= T(I+1)
  120 CONTINUE
C
C        THE ARRAY (D) CONSISTS OF THE SCALED
C        SECOND DIVIDED DIFFERENCES.
C
C        IT IS IMPORTANT THAT WE IDENTIFY DIVIDED DIFFERENCES
C        WHICH ARE ZERO.  THIS MEANS THAT WE MUST COMPARE TWO
C        FLOATING-POINT NUMBERS.  TO DO THIS WE ASSUME D(K) IS
C        ZERO IF D(K) IS SMALL.
C
            XEPS= 1.0
            DO 130 J=1,20
            XEPS= XEPS/10.
            Z= 1.0 + XEPS
            IF ( Z .EQ. 1.0 ) GO TO 135
            YEPS= XEPS
  130       CONTINUE
  135       CONTINUE
            YEPS= YEPS*1000.
C
      DO 140 K=1,N
      D(K)= ( F(K+2)-F(K+1) )/( T(K+2)-T(K+1) )
     C          - ( F(K+1)-F(K) )/( T(K+1)-T(K) )
      IF ( ABS(D(K)) .LE. YEPS ) D(K)= 0.0
  140 CONTINUE
C
C
C        THE INITIAL GUESS (Y) FOR NEWTON'S METHOD
C        WILL YIELD THE SECOND DERIVATIVE OF THE
C        NATURAL SPLINE SOLUTION, EXCEPT POSSIBLY
C        WHEN D(K)= 0.0 FOR SOME K.
C
      DO 145 K=1,N
C
C
      IF ( D(K) .GT. 0.0 ) THEN
            Y(K)=  1.0
      ELSE
            Y(K)= -1.0
      END IF
C
C
  145 CONTINUE
C
C
      WRITE(6,150)
```

```
 150  FORMAT(/,' DATA VALUES ',/)
      WRITE(6,160) (D(I), I=1,N)
 160  FORMAT(5X,4E12.6)
      WRITE(6,170)
 170  FORMAT(//)
C
C
C       ID(K)= 1 INDICATES THAT THE INTERPOLATING FUNCTION
C       IS CONSTRAINED TO BE CONVEX ON [T(K),T(K+1)]
C       AND, HENCE, ITS SECOND DERIVATIVE IS CONSTRAINED
C       TO BE NONNEGATIVE ON THIS INTERVAL.
C
C       ID(K)= -1 INDICATES THAT THE INTERPOLATING FUNCTION
C       IS CONSTRAINED TO BE CONCAVE ON [T(K),T(K+1)]
C       AND, HENCE, ITS SECOND DERIVATIVE IS CONSTRAINED
C       TO BE NONPOSITIVE ON THIS INTERVAL.
C
C       ID(K)= 0 INDICATES THAT THE INTERPOLATING FUNCTION
C       IS UNCONSTRAINED ON [T(K),T(K+1)].
C
C
      DO 180 I=1,N-1
      ID(I+1)= 0
      IF (D(I).GE.0.0 .AND. D(I+1).GE.0.0) ID(I+1)= 1
      IF (D(I).LE.0.0 .AND. D(I+1).LE.0.0) ID(I+1)= -1
 180  CONTINUE
      IF ( D(1) .GE. 0.0 ) THEN
          ID(1)=  1
      ELSE
          ID(1)= -1
      END IF
C
      IF ( D(N) .GE. 0.0 ) THEN
          ID(N+1)=  1
      ELSE
          ID(N+1)= -1
      END IF
C
C       IF A NONZERO DATA VALUE D(I) LIES BETWEEN TWO
C       ZERO DATA VALUES D(I-1) AND D(I+1), THEN D(I)
C       IS TAKEN TO BE ZERO FOR COMPUTATIONAL PURPOSES.
C
      DO 185 I=2,N-1
      IF ( D(I-1).EQ.0.0 .AND. D(I+1).EQ.0.0 ) D(I)= 0.0
 185  CONTINUE
C
C       SUBROUTINE (ZERO) CALCULATES THE PIECEWISE
C       LINEAR SECOND DERIVATIVE OF THE SHAPE-
C       PRESERVING INTERPOLANT.
C
      CALL ZERO(Y,X,N,ITMAX,EPS,IFLAG,TL,TR)
```

```
C
      A(1)=  0.0
      A(M)= 0.0
      DO 190 I=2,N+1
      A(I)= Y(I-1)
  190 CONTINUE
C
      WRITE(6,200)
  200 FORMAT(/,' PIECEWISE LINEAR 2ND DERIVATIVE ',/)
      WRITE(6,210) (I,T(I),A(I), I=1,M)
  210 FORMAT(5X,I5,' ( ',F14.6,' , ',F14.6,' )')
      WRITE(6,220)
  220 FORMAT(//)
C
C       SUBROUTINE (POLY) INTEGRATES THE RESULT
C       FROM SUBROUTINE (ZERO).
C
      CALL POLY(A,T,PP,M,F,LI,TX)
C
      WRITE(6,230)
  230 FORMAT(/,' KNOTS AND COEFFICIENTS OF PIECEWISE CUBIC',/)
      DO 250 I=1,LI
      WRITE(6,240) I,TX(I),(PP(J,I), J=1,4)
  240 FORMAT(5X,I5,5F14.6)
  250 CONTINUE
      WRITE(6,260) IFLAG
  260 FORMAT(/,' ERROR CODE = ',I5,/)
      WRITE(6,270) ITMAX
  270 FORMAT(/,' NUMBER OF ITERATIONS =',I5,)
C
C       SUBROUTINE (DATAFL) IS USED TO CREATE A
C       DATA FILE FOR PLOTTING.  WE EVALUATE THE
C       (JDER)-TH DERIVATIVE OF THE PIECEWISE CUBIC
C       POLYNOMIAL AT MM EQUALLY SPACED POINTS,
C       INCLUDING THE ENDPOINTS TL AND TR. WE
C       ASSUME (JDER) HAS VALUE 0, 1, 2, OR 3.
C
      MM= 201
      JDER= 0
      CALL DATAFL(TX,PP,LI,MM,JDER)
C
      STOP
      END
```

```
      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
```

```
C        OF UNKNOWNS (AND EQUATIONS) REDUCE. IN ORDER
C        TO PERMIT THE COMPUTATION OF ONE JACOBIAN
C        MATRIX THE PROGRAM SETS SUB(K)=SUP(K-1)=0.0
C        AND DIAG(K)=1.0.
C
      DO 125 K=1,N
C
      IF (K.EQ.1) THEN
                    AL= 0.0
                    XL= TL
      ELSE
            AL= A(K-1)
            XL= X(K-1)
      END IF
C
C
      IF (K.EQ.N) THEN
                    AR= 0.0
                    XR= TR
      ELSE
            AR= A(K+1)
            XR= X(K+1)
      END IF
C
C
      IF ( AL.GE.0.0 .AND. A(K).GE.0.0 ) J1= 1
      IF ( AL.LT.0.0 .AND. A(K).GE.0.0 ) J1= 2
      IF ( AL.GE.0.0 .AND. A(K).LT.0.0 ) J1= 3
      IF ( AL.LE.0.0 .AND. A(K).LE.0.0 ) J1= 4
C
      IF ( A(K).GE.0.0 .AND. AR.GE.0.0 ) J2= 1
      IF ( A(K).LT.0.0 .AND. AR.GE.0.0 ) J2= 2
      IF ( A(K).GE.0.0 .AND. AR.LT.0.0 ) J2= 3
      IF ( A(K).LE.0.0 .AND. AR.LE.0.0 ) J2= 4
C
      DT= X(K)-XL
      DA= A(K)-AL
C
      IF ( ID(K) .EQ. 1) THEN
C
          IF (K.NE.1) THEN
C
              IF (J1.EQ.1) THEN
C
      SUB(K)= DT/6.0
      GLEFT=  DT/3.0
C
              ELSE IF (J1.EQ.2) THEN
C
      T= XL-(DT/DA)*AL
      W= 0.5*( X(K)+T )
```

```
      SUB(K)= (X(K)-T)/6.0 * ( ((T-XL)/DT)*((X(K)-T)/DT)
     C              + 4.0*((W-XL)/DT)*((X(K)-W)/DT) )
      GLEFT=  (X(K)-T)/6.0 * ( ((T-XL)/DT)**2
     C              + 4.0*(((W-XL)/DT)**2) + 1.0 )
C
              ELSE IF (J1.EQ.3) THEN
C
      T= XL-(DT/DA)*AL
      W= 0.5*( T+XL )
      SUB(K)= (T-XL)/6.0 * ( 4.0*((W-XL)/DT)*((X(K)-W)/DT)
     C              + ((T-XL)/DT)*((X(K)-T)/DT) )
      GLEFT=  (T-XL)/6.0 * ( 4.0*(((W-XL)/DT)**2)
     C              + ((T-XL)/DT)**2 )
C
              ELSE IF (J1.EQ.4) THEN
C
      SUB(K)= 0.0
      GLEFT=  0.0
C
              END IF
C
          ELSE IF (K.EQ.1) THEN
C
      SUB(1)= 0.0
      GLEFT=  0.0
      IF (J1.EQ.1) GLEFT= DT/3.0
C
          END IF
C
      ELSE IF ( ID(K) .EQ. 0 ) THEN
C
      SUB(K)= DT/6.0
      GLEFT= DT/3.0
C
      ELSE IF ( ID(K) .EQ. -1 ) THEN
C
          IF (K.NE.1) THEN
C
              IF (J1.EQ.4) THEN
C
      SUB(K)= DT/6.0
      GLEFT=  DT/3.0
C
              ELSE IF (J1.EQ.3) THEN
C
      T= XL-(DT/DA)*AL
      W= 0.5*( X(K)+T )
      SUB(K)= (X(K)-T)/6.0 * ( ((T-XL)/DT)*((X(K)-T)/DT)
     C              + 4.0*((W-XL)/DT)*((X(K)-W)/DT) )
      GLEFT=  (X(K)-T)/6.0 * ( ((T-XL)/DT)**2
     C              + 4.0*(((W-XL)/DT)**2) + 1.0 )
```

```
C
                ELSE IF (J1.EQ.2) THEN
C
      T= XL-(DT/DA)*AL
      W= 0.5*( T+XL )
      SUB(K)= (T-XL)/6.0 * ( 4.0*((W-XL)/DT)*((X(K)-W)/DT)
     C                 + ((T-XL)/DT)*((X(K)-T)/DT) )
      GLEFT=  (T-XL)/6.0 * ( 4.0*(((W-XL)/DT)**2)
     C                 + ((T-XL)/DT)**2 )
C
                ELSE IF (J1.EQ.1) THEN
C
      SUB(K)= 0.0
      GLEFT=  0.0
C
                END IF
C
            ELSE IF (K.EQ.1) THEN
C
      SUB(1)= 0.0
      GLEFT=  0.0
      IF (J1.EQ.4) GLEFT= DT/3.0
C
            END IF
C
      END IF
C
      IF (K.NE.1) THEN
      IF ( D(K-1) .EQ. 0.0 ) THEN
           SUB(K) = 0.0
           GLEFT  = 0.0
      END IF
      END IF
C
      DT= XR-X(K)
      DA= AR-A(K)
C
      IF ( ID(K+1) .EQ. 1 ) THEN
C
            IF (K.NE.N) THEN
C
                IF (J2.EQ.1) THEN
C
      SUP(K)= DT/6.0
      GRIGH=  DT/3.0
C
                ELSE IF (J2.EQ.2) THEN
C
      T= X(K)-(DT/DA)*A(K)
      W= 0.5*( XR+T )
      SUP(K)= (XR-T)/6.0 * ( ((T-X(K))/DT)*((XR-T)/DT)
```

```
     C              + 4.0*((W-X(K))/DT)*((XR-W)/DT) )
      GRIGH=  (XR-T)/6.0 * ( ((XR-T)/DT)**2
     C              + 4.0*(((XR-W)/DT)**2) )
C
                ELSE IF (J2.EQ.3) THEN
C
      T= X(K)-(DT/DA)*A(K)
      W= 0.5*( T+X(K) )
      SUP(K)= (T-X(K))/6.0 * ( 4.0*((W-X(K))/DT)*((XR-W)/DT)
     C              + ((T-X(K))/DT)*((XR-T)/DT) )
      GRIGH=  (T-X(K))/6.0 * ( 1.0 + 4.0*(((XR-W)/DT)**2)
     C              + ((XR-T)/DT)**2 )
C
                ELSE IF (J2.EQ.4) THEN
C
      SUP(K)= 0.0
      GRIGH=  0.0
C
                END IF
C
            ELSE IF (K.EQ.N) THEN
C
      SUP(N)= 0.0
      GRIGH=  0.0
      IF (J2.EQ.1) GRIGH= DT/3.0
C
            END IF
C
      ELSE IF ( ID(K+1) .EQ. 0 ) THEN
C
      SUP(K)= DT/6.0
      GRIGH=  DT/3.0
C
      ELSE IF ( ID(K+1) .EQ. -1 ) THEN
C
            IF (K.NE.N) THEN
C
                IF (J2.EQ.4) THEN
C
      SUP(K)= DT/6.0
      GRIGH=  DT/3.0
C
                ELSE IF (J2.EQ.3) THEN
C
      T= X(K)-(DT/DA)*A(K)
      W= 0.5*( XR+T )
      SUP(K)= (XR-T)/6.0 * ( ((T-X(K))/DT)*((XR-T)/DT)
     C              + 4.0*((W-X(K))/DT)*((XR-W)/DT) )
      GRIGH=  (XR-T)/6.0 * ( ((XR-T)/DT)**2
     C              + 4.0*(((XR-W)/DT)**2) )
C
```

```
                  ELSE IF (J2.EQ.2) THEN
C
          T= X(K)-(DT/DA)*A(K)
          W= 0.5*( T+X(K) )
          SUP(K)= (T-X(K))/6.0 * ( 4.0*((W-X(K))/DT)*((XR-W)/DT)
     C                   + ((T-X(K))/DT)*((XR-T)/DT) )
          GRIGH=  (T-X(K))/6.0 * ( 1.0 + 4.0*(((XR-W)/DT)**2)
     C                   + ((XR-T)/DT)**2 )
C
                  ELSE IF (J2.EQ.1) THEN
C
          SUP(K)= 0.0
          GRIGH=  0.0
C
                  END IF
C
              ELSE IF (K.EQ.N) THEN
C
          SUP(N)= 0.0
          GRIGH= 0.0
          IF (J2.EQ.4) GRIGH= DT/3.0
C
              END IF
C
          END IF
C
          IF (K.NE.N) THEN
          IF ( D(K+1) .EQ. 0.0 ) THEN
               SUP(K)= 0.0
               GRIGH = 0.0
          END IF
          END IF
C
          DIAG(K)= GLEFT+GRIGH
C
C
          IF ( D(K) .EQ. 0.0 ) THEN
                DIAG(K)= 1.0
                SUB(K)=  0.0
                SUP(K)=  0.0
          END IF
C
  125     CONTINUE
C
          DO 150 L=1,N
          H(L)= D(L)
  150     CONTINUE
C
C
            WE SOLVE THE MATRIX EQUATION JX=H, THE ARRAY (H)
```

```
C         BEING IDENTICAL TO THE ARRAY (D).  THE SOLUTION
C         IS RETURNED IN THE ARRAY (H).
C
C
          CALL TRID(SUB,DIAG,SUP,H,N)
C
          SUM1= 0.0
          DO 200 L=1,N
          A(L)= H(L)
          SUM1= SUM1 + ABS(A(L))
      200 CONTINUE
C
C         THE FUNCTION EVALUATION SUBROUTINE COMPUT MAY
C         BE DELETED. IN THIS CASE THE FOLLOWING EIGHT
C         LINES ARE TO BE DELETED AND THE ARRAY (FX)
C         CAN BE TAKEN FROM THE REAL STATEMENT AT THE
C         BEGINNING OF THIS SUBROUTINE.
C
          CALL COMPUT(A,FX,N,X,TL,TR)
          FNORM1= 0.0
          DO 250 L=1,N
          FNORM1= FNORM1 + FX(L)*FX(L)
      250 CONTINUE
          FNORM1= SQRT(FNORM1)
          WRITE(6,300) LJ,FNORM1
      300 FORMAT(I5,E15.6)
C
C
          IF (LJ.NE.1) THEN
          RATIO= ABS(SUM1-SUM2)
          AB= EPS*SUM2
          IFLAG= 1
          IF (RATIO .LE. AB) GO TO 400
          END IF
          SUM2= SUM1
      350 CONTINUE
          IFLAG= 2
      400 CONTINUE
          ITMAX= LJ
          RETURN
          END
```

```
      SUBROUTINE COMPUT(A,FX,N,X,TL,TR)
C
C       SUBROUTINE (COMPUT), THE FUNCTION EVAULATING
C       SUBROUTINE, IS OPTIONAL.
C
C
      REAL A(N),FX(N),F1,ALO,AHI,TLO,THI
      REAL GLEF,GRIG,TS,X(N)
      INTEGER N,K,J1,J2
      COMMON D(50),ID(50)
      DO 100 K=1,N
C
      IF ( D(K) .NE. 0.0 ) THEN
C
      IF (K.EQ.1) THEN
                  ALO= 0.0
                  TLO= TL
      ELSE
                  ALO= A(K-1)
                  TLO= X(K-1)
      END IF
C
      IF (K.EQ.N) THEN
                  AHI= 0.0
                  THI= TR
      ELSE
                  AHI= A(K+1)
                  THI= X(K+1)
      END IF
C
C
C
      IF (ALO.GE.0.0 .AND. A(K).GE.0.0) J1= 1
      IF (ALO.LT.0.0 .AND. A(K).GE.0.0) J1= 2
      IF (ALO.GE.0.0 .AND. A(K).LT.0.0) J1= 3
      IF (ALO.LT.0.0 .AND. A(K).LT.0.0) J1= 4
C
      IF (A(K).GE.0.0 .AND. AHI.GE.0.0) J2= 1
      IF (A(K).LT.0.0 .AND. AHI.GE.0.0) J2= 2
      IF (A(K).GE.0.0 .AND. AHI.LT.0.0) J2= 3
      IF (A(K).LT.0.0 .AND. AHI.LT.0.0) J2= 4
C
      DT= X(K)-TLO
C
          IF ( ID(K) .EQ. 1 ) THEN
C
              IF (J1.EQ.1) THEN
C
      GLEF= DT*( 2.0*A(K) + ALO )/6.0
C
```

```
                    ELSE IF (J1.EQ.2) THEN
C
            TS= TLO - ALO*DT/( A(K)-ALO )
            F1= ( (TS+X(K))*0.5 - TLO )/DT
            GLEF= (X(K)-TS)*A(K)*( 2.0*F1 + 1.0 )/6.0
C
                    ELSE IF (J1.EQ.3) THEN
C
            TS= TLO - ALO*DT/( A(K)-ALO )
            GLEF= ( (TS-TLO)**2 )*ALO/(6.0*DT)
C
                    ELSE IF (J1.EQ.4) THEN
C
            GLEF= 0.0
C
                    END IF
C
                ELSE IF ( ID(K) .EQ. 0 ) THEN
C
            GLEF= DT*( 2.0*A(K) + ALO )/6.0
C
                ELSE IF ( ID(K) .EQ. -1 ) THEN
C
                    IF (J1.EQ.4) THEN
C
            GLEF= DT*( 2.0*A(K) + ALO )/6.0
C
                    ELSE IF (J1.EQ.3) THEN
C
            TS= TLO - ALO*DT/( A(K)-ALO )
            F1= ( (TS+X(K))*0.5 - TLO )/DT
            GLEF= (X(K)-TS)*A(K)*( 2.0*F1 + 1.0 )/6.0
C
                    ELSE IF (J1.EQ.2) THEN
C
            TS= TLO - ALO*DT/( A(K)-ALO )
            GLEF= ( (TS-TLO)**2 )*ALO/(6.0*DT)
C
                    ELSE IF (J1.EQ.1) THEN
C
            GLEF= 0.0
C
                    END IF
C
                END IF
C
            DT= THI-X(K)
C
                IF ( ID(K+1) .EQ. 1 ) THEN
C
                    IF (J2.EQ.1) THEN
```

```
C
      GRIG= DT*( 2.0*A(K) + AHI )/6.0
C
          ELSE IF (J2.EQ.2) THEN
C
      TS= X(K) - A(K)*DT/( AHI-A(K) )
      GRIG= ( (THI-TS)**2 )*AHI/(6.0*DT)
C
          ELSE IF (J2.EQ.3) THEN
C
      TS= X(K) - A(K)*DT/( AHI-A(K) )
      F1= ( THI-0.5*( TS+X(K) ) )/DT
      GRIG= ( TS-X(K) )*A(K)*( 1.0 + 2.0*F1 )/6.0
C
          ELSE IF (J2.EQ.4) THEN
C
      GRIG= 0.0
C
          END IF
C
        ELSE IF ( ID(K+1) .EQ. 0 ) THEN
C
      GRIG= DT*( 2.0*A(K) + AHI )/6.0
C
        ELSE IF ( ID(K+1) .EQ. -1 ) THEN
C
          IF (J2.EQ.4) THEN
C
      GRIG= DT*( 2.0*A(K) + AHI )/6.0
C
          ELSE IF (J2.EQ.3) THEN
C
      TS= X(K) - A(K)*DT/( AHI-A(K) )
      GRIG= ( (THI-TS)**2 )*AHI/(6.0*DT)
C
          ELSE IF (J2.EQ.2) THEN
C
      TS= X(K) - A(K)*DT/( AHI-A(K) )
      F1= ( THI-0.5*( TS+X(K) ) )/DT
      GRIG= ( TS-X(K) )*A(K)*( 1.0 + 2.0*F1 )/6.0
C
          ELSE IF (J2.EQ.1) THEN
C
      GRIG= 0.0
C
          END IF
C
        END IF
C
C
        IF (K.NE.1) THEN
```

```
              IF ( D(K-1) .EQ. 0.0 ) GLEF= 0.0
          END IF
C
          IF (K.NE.N) THEN
              IF ( D(K+1) .EQ. 0.0 ) GRIG= 0.0
          END IF
C
      FX(K)= GLEF + GRIG - D(K)
C
      ELSE IF ( D(K) .EQ. 0.0 ) THEN
                                    FX(K)= 0.0
      END IF
C
100   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE POLY(A,T,PP,M,F,LI,TX)
C
C          SUBROUTINE POLY INTEGRATES BACK TWICE THE
C       POSITIVE PART OF THE PIECEWISE LINEAR SECOND
C       DERIVATIVE WHERE THE DATA SUGGESTS THAT THE
C       INTERPOLATING CURVE SHOULD BE CONVEX, THE
C       NEGATIVE PART OF THE PIECEWISE LINEAR SECOND
C       DERIVATIVE WHERE THE DATA SUGGESTS THAT THE
C       INTERPOLATING CURVE SHOULD BE CONCAVE, AND
C       THE REMAINING PORTION OF THE PIECEWISE LINEAR
C       SECOND DERIVATIVE ON THE TRANSITION INTERVALS.
C
C          THE INTEGRATION YIELDS A PIECEWISE CUBIC
C       POLYNOMIAL WITH KNOTS GIVEN BY THE SEQUENCE
C       (TX).  THIS CUBIC POLYNOMIAL INTERPOLATES THE
C       DATA AND ITS COEFFICIENTS ARE DENOTED BY THE
C       NUMBERS PP(J,I) - THE VALUE OF THE (J-1)ST
C       DERIVATIVE OF THE FUNCTION EVALUATED AT TX(I).
C       FOR X SUCH THAT TX(I).GE.X.LT.TX(I+1) THE VALUE
C       OF THE CUBIC POLYNOMIAL IS
C
C                 PP(1,I)
C        +        PP(2,I) * ( X-TX(I) )
C        + (1/2)PP(3,I) * ( X-TX(I) )**2
C        + (1/6)PP(4,I) * ( X-TX(I) )**3
C
C
      INTEGER M,J,L,LI
      REAL A(50),T(50),PP(4,100),F(50),TX(100),TAU
      REAL DF,DT,DA,C,E
      COMMON D(50),ID(50)
      LI= 1
      MN1= M-1
      DO 100 L=1,MN1
      DF= F(L+1)-F(L)
      DT= T(L+1)-T(L)
      DA= A(L+1)-A(L)
C
      JP= 0
      IF (L.EQ.1) THEN
                  IF ( D(1) .EQ. 0.0 ) JP= 1
      ELSE IF (L.EQ.MN1) THEN
                  IF ( D(M-2) .EQ. 0.0 ) JP= 1
      ELSE
                  C= D(L-1)*D(L)
                  IF ( C .EQ. 0.0 ) JP= 1
      END IF
C
C
C
```

```fortran
      IF (JP.EQ.1) THEN
C
      PP(4,LI)= 0.0
      PP(3,LI)= 0.0
      PP(2,LI)= DF/DT
      PP(1,LI)= F(L)
      TX(LI)= T(L)
      LI= LI+1
C
      ELSE IF (JP.EQ.0) THEN
C
      IF (A(L).GE.0.0 .AND. A(L+1).GE.0.0) J= 1
      IF (A(L).LT.0.0 .AND. A(L+1).GT.0.0) J= 2
      IF (A(L).GT.0.0 .AND. A(L+1).LT.0.0) J= 3
      IF (A(L).LE.0.0 .AND. A(L+1).LE.0.0) J= 4
C
          IF ( ID(L) .EQ. 1) THEN
C
              IF (J.EQ.1) THEN
C
      C= DF/DT - (DA/6.0 + A(L)/2.0)*DT
      PP(4,LI)= DA/DT
      PP(3,LI)= A(L)
      PP(2,LI)= C
      PP(1,LI)= F(L)
      TX(LI)= T(L)
      LI= LI+1
C
              ELSE IF (J.EQ.2) THEN
C
      TAU= T(L) - A(L)*DT/DA
      C= DF/DT - (A(L+1)**3)*DT/(6.0*DA*DA)
      PP(4,LI)= 0.0
      PP(3,LI)= 0.0
      PP(2,LI)= C
      PP(1,LI)= F(L)
      PP(4,LI+1)= DA/DT
      PP(3,LI+1)= 0.0
      PP(2,LI+1)= C
      PP(1,LI+1)= C*(TAU-T(L)) + F(L)
      TX(LI)= T(L)
      TX(LI+1)= TAU
      LI= LI+2
C
              ELSE IF (J.EQ.3) THEN
C
      TAU= T(L) - A(L)*DT/DA
      E= F(L) - (A(L)**3)*DT*DT/(6.0*DA*DA)
      C= DF/DT + (A(L)**3)*DT/(6.0*DA*DA)
      PP(4,LI)= DA/DT
      PP(3,LI)= A(L)
```

```
      PP(2,LI)= C + A(L)*A(L)*DT*0.5/DA
      PP(1,LI)= F(L)
      PP(4,LI+1)= 0.0
      PP(3,LI+1)= 0.0
      PP(2,LI+1)= C
      PP(1,LI+1)= C*( TAU-T(L) ) + E
      TX(LI)= T(L)
      TX(LI+1)= TAU
      LI= LI+2
C
              ELSE IF (J.EQ.4) THEN
C
      PP(4,LI)= 0.0
      PP(3,LI)= 0.0
      PP(2,LI)= DF/DT
      PP(1,LI)= F(L)
      TX(LI)= T(L)
      LI= LI+1
C
              END IF
C
          ELSE IF ( ID(L) .EQ. 0 ) THEN
C
      C= DF/DT - (DA/6.0 + A(L)/2.0)*DT
      PP(4,LI)= DA/DT
      PP(3,LI)= A(L)
      PP(2,LI)= C
      PP(1,LI)= F(L)
      TX(LI)= T(L)
      LI= LI+1
C
          ELSE IF ( ID(L) .EQ. -1 ) THEN
C
              IF (J.EQ.4) THEN
C
      C= DF/DT - (DA/6.0 + A(L)/2.0)*DT
      PP(4,LI)= DA/DT
      PP(3,LI)= A(L)
      PP(2,LI)= C
      PP(1,LI)= F(L)
      TX(LI)= T(L)
      LI= LI+1
C
              ELSE IF (J.EQ.3) THEN
C
      TAU= T(L) - A(L)*DT/DA
      C= DF/DT - (A(L+1)**3)*DT/(6.0*DA*DA)
      PP(4,LI)= 0.0
      PP(3,LI)= 0.0
      PP(2,LI)= C
      PP(1,LI)= F(L)
```

```
      PP(4,LI+1)= DA/DT
      PP(3,LI+1)= 0.0
      PP(2,LI+1)= C
      PP(1,LI+1)= C*(TAU-T(L)) + F(L)
      TX(LI)= T(L)
      TX(LI+1)= TAU
      LI= LI+2
C
              ELSE IF (J.EQ.2) THEN
C
      TAU= T(L) - A(L)*DT/DA
      E= F(L) - (A(L)**3)*DT*DT/(6.0*DA*DA)
      C= DF/DT + (A(L)**3)*DT/(6.0*DA*DA)
      PP(4,LI)= DA/DT
      PP(3,LI)= A(L)
      PP(2,LI)= C + A(L)*A(L)*DT*0.5/DA
      PP(1,LI)= F(L)
      PP(4,LI+1)= 0.0
      PP(3,LI+1)= 0.0
      PP(2,LI+1)= C
      PP(1,LI+1)= C*( TAU-T(L) ) + E
      TX(LI)= T(L)
      TX(LI+1)= TAU
      LI= LI+2
C
              ELSE IF (J.EQ.1) THEN
C
      PP(4,LI)= 0.0
      PP(3,LI)= 0.0
      PP(2,LI)= DF/DT
      PP(1,LI)= F(L)
      TX(LI)= T(L)
      LI= LI+1
C
              END IF
C
          END IF
C
      END IF
C
  100 CONTINUE
      PP(4,LI)= 0.0
      PP(3,LI)= 0.0
      PP(2,LI)= 0.0
      PP(1,LI)= F(M)
      TX(LI)= T(M)
      RETURN
      END
```

Subroutines TRID and DATAFL are listed in Appendix A.

1 Report No
NASA CR-178001

2 Government Accession No

3 Recipient's Catalog No

4 Title and Subtitle
Minimal Norm Constrained Interpolation

5 Report Date
September 1985

6 Performing Organization Code

7 Author(s)
Larry Dean Irvine

8 Performing Organization Report No

10 Work Unit No

9 Performing Organization Name and Address
Old Dominion University
Norfolk, Virginia

11 Contract or Grant No
NAG-1-363

13 Type of Report and Period Covered
Contractor Report

12 Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546

14 Sponsoring Agency Code
505-31-83-02

15 Supplementary Notes
This report is a dissertation submitted to Old Dominion University for partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computational and Applied Mathematics. Langley Technical Monitor: Robert E. Smith, Jr.

16 Abstract

In computational fluid dynamics and in CAD/CAM, a physical boundary is usually known only discreetly and most often must be approximated. An acceptable approximation preserves the salient features of the data such as convexity and concavity. In this dissertation, a smooth interpolant which is locally concave where the data are concave and is locally convex where the data are convex is described. The interpolant is found by posing and solving a minimization problem whose solution is a piecewise cubic polynomial. The problem is solved indirectly by using the Peano Kernal theorem to recast it into an equivalent minimization problem having the second derivative of the interpolant as the solution. This approach leads to the solution of a nonlinear system of equations. It is shown that Newton's method is an exceptionally attractive and efficient method for solving the nonlinear system of equations. Examples of shape-preserving interpolants as well as convergence results obtained by using Newton's method are also shown. A FORTRAN program to compute these interpolants is listed. The problem of computing the interpolant of minimal norm from a convex cone in a normal dual space is also discussed. An extension of de Boor's work on minimal norm unconstrained interpolation is presented.

17 Key Words (Suggested by Author(s))
Surface definition, Interpolation,
Spline functions

18 Distribution Statement
Unclassified -- Unlimited

Subject Category - 64

19 Security Classif (of this report)	20 Security Classif (of this page)	21 No of Pages	22 Price
UNCLASSIFIED	UNCLASSIFIED	108	A06

End of Document