ALGORITHM 507
Procedures for Quintic Natural Spline Interpolation [E1]

JOHN G. HERRIOT
Stanford University
and
CHRISTIAN H. REINSCH
Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften, Germany

Key Words and Phrases: approximation, interpolation, spline, spline approximation, quintic natural spline
CR Categories: 5.13
Language: Algol W

DESCRIPTION

1. Introduction

The purpose of the procedures presented here is to determine the interpolating quintic natural spline function $S(x)$ for the set of data points (x_i, y_i), $i = N1, N1+1, \ldots, N2$, where it is assumed that $x_{N1} < x_{N1+1} < \cdots < x_{N2}$. The interpolating quintic natural spline function $S(x)$ with the knots $x_{N1}, \ldots, x_{N2}$ has the following properties: (i) $S(x)$ is a polynomial of degree 5 in each interval (x_i, x_{i+1}), $i = N1, \ldots, N2-1$. (ii) $S(x)$ and its derivatives $S'(x)$, $S''(x)$, $S'''(x)$, and $S''''(x)$ are continuous in $[x_{N1}, x_{N2}]$. (iii) $S'''(x_{N1}) = S'''(x_{N2}) = S''''(x_{N1}) = S''''(x_{N2}) = 0$. (iv) $S(x_i) = y_i$, $i = N1, \ldots, N_2$. It is known that if $N2 > N1+1$, then there is a unique quintic natural spline function which has the properties (i)–(iv). (See, for example, Greville [3, 4].) This spline function can be represented in the form

$$S(x) = y_i + B_i t + C_i t^2 + D_i t^3 + E_i t^4 + F_i t^5 \qquad (1)$$

with $t = x - x_i$ for $x_i \le x < x_{i+1}$, $i = N1, \ldots, N2-1$.

The procedure *QUINAT* computes the coefficients B_i, C_i, D_i, E_i, F_i of the quintic natural spline represented as in eq. (1) for an arbitrary set of data points

---

Received 20 June 1974.

This work was supported in part by the National Science Foundation, under Grant GJ-29988X.
Authors' addresses: J.G Herriot, Department of Computer Science, Stanford University Stanford, CA 94305; C.H. Reinsch, Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften, 8 Munchen 2, Germany.

(x_i, y_i) as previously specified. This procedure is much faster than the procedure *NATSPLINE* of ACM Algorithm 472 [6] with $m = 3$. An even faster procedure, *QUINEQ*, is provided for the case in which the knots x_i are known to be equidistant. In this case it is not necessary to specify the values of x_i. The representation (1) is still used, but now $t = (x-x_i)/h$, where $h = x_{i+1}-x_i$, the constant spacing of the knots.

If at one or more of the knots x_i one also specifies the derivative y_i', thus requiring $S'(x_i) = y_i'$, then one has to give up the condition that $S''''(x)$ be continuous at the knot x_i. If the second derivative y_i'' is also specified, thus requiring $S''(x_i) = y_i''$, then one must also give up the condition that $S'''(x)$ be continuous at x_i. *QUINAT* is designed so that it can be used in these cases with the convention that if two consecutive knots are equal, say $x_j = x_{j+1}$, then $S(x_j) = y_j$ and $S'(x_j) = y_{j+1}$, and if three consecutive knots are equal, say $x_j = x_{j+1} = x_{j+2}$, then $S(x_j) = y_j$, $S'(x_j) = y_j$, and $S''(x_j) = y_{j+2}$. Thus in order to use *QUINAT* in the case that both the value y_j and the first derivative y_j' are specified at x_j, one increases the number of knots by 1 setting $x_{j+1} = x_j$ (and renumbering the knots and values to the right). Then one chooses $y_{j+1} = y_j'$. The spline function computed by *QUINAT* will have the property $S(x_j) = y_j$ and $S'(x_j) = y_{j+1}$. One may use *QUINAT* in a similar manner if the second derivative is also specified at a knot x_j. Complete details are given in the comment of the procedure *QUINAT*.

If the values of the function y_i and the values of the first derivative y_i' are specified at all the knots x_i, then $S''''(x)$ need not be continuous at the knots and $S''''(x_{N1})$ and $S''''(x_{N2})$ need not be zero. Such a spline is said to be of deficiency 2. The procedure *QUINDF* computes the coefficients of the quintic natural spline of deficiency 2 when the values of the function y_i and the values of the first derivative y_i' are given at each knot. Although *QUINAT* could be used for this case as just described, *QUINDF* is much faster and needs much less storage space.

It is not of interest to specify the values of the function and its first and second derivatives at each knot, because in this case the quintic polynomial is completely determined in each interval independently of all other intervals.

2. Method of Calculation

QUINAT. As in the general case of Algorithm 472 [6], the calculation of the coefficients of the spline function is carried out in a numerically stable manner following a method described by Anselone and Laurent [1]. The basic ideas on which the method is based were given earlier by Schoenberg [7]. The method is specialized to the case of the quintic natural spline and uses minimum support B-splines [2, 3] of degree 2 to form a basis for the class of third derivatives of the quintic natural splines. Instead of specializing the formulas of Algorithm 472 [6] by setting $m = 3$, we derive the necessary formulas directly and we choose a different numbering and a different normalization for the B-splines.

We first assume that the knots are strictly monotone increasing, that is, $x_{N1} < x_{N1+1} < \cdots < x_{N2}$. In order to simplify the notation, we choose $N1 = 0$ and let $N2 = n$, so that the data points are denoted by (x_i, y_i), $i = 0, 1, \ldots, n$. We denote by $M_i(x)$ the B-spline of degree 2, vanishing outside the interval (x_{i-1}, x_{i+2}). We let $h_i = x_{i+1} - x_i$, $t = x - x_{i-1}$, $u = x - x_i$, $v = x - x_{i+1}$.

Then we have

$$
\begin{aligned}
M_i(x) &= At^2, & x_{i-1} \le x < x_i\,, \\
&= B + Cu - Du^2, & x_i \le x < x_{i+1}\,, \\
&= E(v - h_{i+1})^2, & x_{i+1} \le x < x_{i+2}\,,
\end{aligned} \tag{2}
$$

where

$$
\begin{aligned}
&A = 1/[h_{i-1}(h_{i-1} + h_i)], \quad B = h_{i-1}/(h_{i-1} + h_i), \quad C = 2/(h_{i-1} + h_i), \\
&D = (h_{i-1} + 2h_i + h_{i+1})/[(h_{i-1} + h_i)h_i(h_i + h_{i+1})], \quad E = 1/[h_{i+1}(h_i + h_{i+1})].
\end{aligned} \tag{3}
$$

Now since the third derivative $S'''(x)$ vanishes outside the interval (x_0, x_n), it has a unique representation of the form

$$
S'''(x) = \sum_{j=1}^{n-2} 60\gamma_j M_j(x). \tag{4}
$$

In order to determine the γ_j, we make use of the relation

$$
\int_{-\infty}^{\infty} M_i(x) S'''(x)dx = 2(S(x_i, x_{i+1}, x_{i+2}) - S(x_{i-1}, x_i, x_{i+1})) \tag{5}
$$

using the usual notation for divided differences. This relation is easily obtained by integration by parts. If we multiply eq. (4) by $\frac{1}{2}M_i(x)$, $i = 1, 2, \dots, n-2$, and integrate, we obtain a well-conditioned positive definite pentadiagonal system of linear equations for the determination of the γ_j:

$$
\begin{aligned}
d_1\gamma_1 + e_1\gamma_2 + f_1\gamma_3 &= c_1 \\
e_1\gamma_1 + d_2\gamma_2 + e_2\gamma_3 + f_2\gamma_4 &= c_2 \\
f_{i-2}\gamma_{i-2} + e_{i-1}\gamma_{i-1} + d_i\gamma_i + e_i\gamma_{i+1} + f_i\gamma_{i+2} &= c_i\,, \quad i = 3, 4, \dots, n-4 \\
f_{n-5}\gamma_{n-5} + e_{n-4}\gamma_{n-4} + d_{n-3}\gamma_{n-3} + e_{n-3}\gamma_{n-2} &= c_{n-3} \\
f_{n-4}\gamma_{n-4} + e_{n-3}\gamma_{n-3} + d_{n-2}\gamma_{n-2} &= c_{n-2}
\end{aligned} \tag{6}
$$

where

$$
\begin{aligned}
d_i &= T_1 + T_2 + T_3\,, & i &= 1,2,\dots,n-2, \\
e_i &= T_4 + T_5\,, & i &= 1,2,\dots,n-3, \\
f_i &= T_6\,, & i &= 1,2,\dots,n-4, \\
c_i &= y_{i,i+1,i+2} - y_{i-1,i,i+1}\,, & i &= 1,2,\dots,n-2.
\end{aligned}
$$

Here $y_{i,i+1,i+2}$ denotes the second divided difference of the given $\{y_i\}$, and for the T_i one finds, after some algebraic manipulation, the following formulas:

$$
\begin{aligned}
T_1 &= 6h_{i-1}^3/(h_{i-1} + h_i)^2 \\
T_2 &= h_i\{30h_{i-1}^2h_{i+1}^2 + (h_{i-1} + h_{i+1})h_i(40h_{i-1}h_{i+1} + 14h_i^2) \\
&\qquad + h_i^2[16(h_{i-1}^2 + h_{i+1}^2) + 42h_{i-1}h_{i+1} + 4h_i^2]\}/[(h_{i-1} + h_i)^2(h_i + h_{i+1})^2] \\
T_3 &= 6h_{i+1}^3/(h_i + h_{i+1})^2 \\
T_4 &= h_i^2[h_{i-1}(h_i + h_{i+1}) + 3(h_{i-1} + h_i)(h_i + 3h_{i+1})]/[(h_{i-1} + h_i)(h_i + h_{i+1})^2] \\
T_5 &= h_{i+1}^2[h_{i+2}(h_i + h_{i+1}) + 3(h_{i+1} + h_{i+2})(3h_i + h_{i+1})]/[(h_i + h_{i+1})^2(h_{i+1} + h_{i+2})] \\
T_6 &= h_{i+1}^3/[(h_i + h_{i+1})(h_{i+1} + h_{i+2})].
\end{aligned}
$$

Note that all terms in these expressions are positive; consequently no cancellations can occur. The system of equations (6) can be solved for the γ_j by using Gaussian elimination without pivoting. When the coefficients γ_j have been found, $S'''(x)$ is given by eq. (4). Remembering that $M_j(x)$ vanishes outside the interval (x_{j-1}, x_{j+2}) and making use of eqs. (1) through (4), we easily find that

$$\left.\begin{aligned} D_i/10 &= (\gamma_{i-1}h_i + \gamma_i h_{i-1})/(h_{i-1} + h_i) \\ E_i/5 &= (\gamma_i - \gamma_{i-1})/(h_{i-1} + h_i) \\ F_i &= (1/h_i)[(\gamma_{i+1} - \gamma_i)/(h_i + h_{i+1}) \\ &\quad - (\gamma_i - \gamma_{i-1})/(h_{i-1} + h_i)]. \end{aligned}\right\} \quad i = 2, 3, \ldots, n-3.$$

These formulas can also be used for $i = 0, 1, n-2, n-1$ by adding the convention that $\gamma_{-1} = \gamma_0 = \gamma_{n-1} = \gamma_n = 0$. (Note that $D_0 = E_0 = 0$ as they should.) Finally we make use of the continuity of $S(x)$ and its first four derivatives at x_i to obtain the following formulas for B_i and C_i :

$$B_i = \frac{h_{i-1}}{h_{i-1} + h_i}\,\frac{y_{i+1} - y_i}{h_i} + \frac{h_i}{h_{i-1} + h_i}\,\frac{y_i - y_{i-1}}{h_{i-1}} - D_i h_{i-1} h_i + E_i h_{i-1} h_i (h_{i-1} - h_i) - \frac{h_{i-1} h_i}{h_{i-1} + h_i} (F_{i-1} h_{i-1}^3 + F_i h_i^3)$$

$$C_i = \frac{1}{h_{i-1} + h_i}\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right) + D_i (h_{i-1} - h_i) - E_i \frac{h_{i-1}^3 + h_i^3}{h_{i-1} + h_i} + \frac{1}{h_{i-1} + h_i} (F_{i-1} h_{i-1}^4 - F_i h_i^4).$$

These formulas are valid for $i = 1, 2, \ldots, n-1$. In addition, we have for the endpoints:

$$C_0 = C_1 - 10F_0\, h_0^3, \qquad B_0 = (y_1 - y_0)/h_0 - C_0\, h_0 - F_0\, h_0^4,$$
$$C_n = C_{n-1} + 10F_{n-1}h_{n-1}^3, \qquad B_n = (y_n - y_{n-1})/h_{n-1} + C_n h_{n-1} - F_{n-1}\, h_{n-1}^4.$$

In the preceding discussion we have assumed that the knots were distinct. We can relax this condition and allow two or three consecutive knots to be equal. The procedure *QUINAT* has been written in such a way that if $x_j = x_{j+1}$, then $S(x_j) = y_j$ and $S'(x_j) = y_{j+1}$, and if $x_j = x_{j+1} = x_{j+2}$, then, in addition, $S''(x_j) = y_{j+2}$. The use of *QUINAT* in these cases is fully explained in its comment.

QUINEQ. The calculation of the coefficients in *QUINEQ* for the case of equidistant knots is carried out in the same manner as is the calculation of the coefficients in *QUINAT* for the general case. However, there are a number of simplifications which result in considerable economy of computational effort. It is not necessary to specify x_i. Hence we can assume $x_i = i$. Then $h_i = 1$ for all i, and the coefficients of $M_i(x)$ are independent of i as are also the d_i, e_i, f_i of the pentadiagonal system (6) for the γ_i. Thus eqs. (2) reduce to

$$M_i(x) = \begin{cases} \frac{1}{2}\, t^2, & i - 1 \le x < i, \\ \frac{1}{2} + u - u^2, & i \le x < i + 1, \\ \frac{1}{2}\,(v - 1)^2, & i + 1 \le x < i + 2, \end{cases}$$

with $t = x - (i - 1)$, $u = x - i$, $v = x - (i + 1)$.

Instead of eq. (4) it is convenient to take

$$S'''(x) = \sum_{j=0}^{n-3} 120\gamma_j M_{j+1}(x).$$

The divided differences become ordinary differences so that eq. (5) becomes:

$$\int_{-\infty}^{\infty} M_i(x) S'''(x)\,dx = \Delta^3 S(x_{i-1}).$$

The pentadiagonal system (6) for the determination of γ_j becomes:

$$\begin{aligned}
66\gamma_0 + 26\gamma_1 + \gamma_2 &= \Delta^3 y_0 \\
26\gamma_0 + 66\gamma_1 + 26\gamma_2 + \gamma_3 &= \Delta^3 y_1 \\
\gamma_{i-2} + 26\gamma_{i-1} + 66\gamma_i + 26\gamma_{i+1} + \gamma_{i+2} &= \Delta^3 y_i\,, \quad i = 2, 3, \ldots, n-5 \\
\gamma_{n-6} + 26\gamma_{n-5} + 66\gamma_{n-4} + 26\gamma_{n-3} &= \Delta^3 y_{n-4} \\
\gamma_{n-5} + 26\gamma_{n-4} + 66\gamma_{n-3} &= \Delta^3 y_{n-3}\,.
\end{aligned}$$

The equations for the determination of the spline function coefficients then become:

$$\begin{aligned}
D_i/10 &= \gamma_{i-2} + \gamma_{i-1} & B_i &= \tfrac{1}{2}(y_{i+1} - y_{i-1} - F_{i-1} - F_i) - D_i \\
E_i/5 &= \gamma_{i-1} - \gamma_{i-2} & C_i &= \tfrac{1}{2}(y_{i+1} + y_{i-1} + F_{i-1} - F_i) - y_i - E_i\,. \\
F_i &= \gamma_i - \gamma_{i-1} - \gamma_{i-1} + \gamma_{i-2}
\end{aligned}$$

These formulas are valid for $i = 1, 2, \ldots, n-1$ with the convention that $\gamma_{-1} = \gamma_{n-2} = \gamma_{n-1} = 0$. The formula for F_i can be used for $i = 0$ by setting $\gamma_{-2} = 0$. (Note that $D_0 = E_0 = 0$ as they should.) Finally the coefficients B_i and C_i at the endpoints are given by

$$\begin{aligned}
C_0 &= C_1 - 10F_0, & B_0 &= y_1 - y_0 - C_0 - F_0, \\
C_n &= C_{n+1} + 10F_{n-1}, & B_n &= y_n - y_{n-1} + C_n - F_{n-1}\,.
\end{aligned}$$

QUINDF. We now assume $S(x_i) = y_i$ and $S'(x_i) = y_i'$ are specified at each of the knots. We must exclude the possibility that $x_i = x_{i+1}$ as this would imply a multiplicity of 4, which is not feasible for quintic splines.

We could proceed as in the calculation of *QUINAT* by using minimum support B-splines of degree 2 to form a basis for the class of third derivatives of the quintic natural splines. Of course, the B-splines would also have to be of deficiency 2. We would again obtain a pentadiagonal system of equations which could be solved and then the coefficients for the deficient quintic natural spline could be calculated. An algorithm based on this method was developed and tested by the present authors [5].

However, we have found that we can obtain a more efficient algorithm by imposing the appropriate continuity conditions directly on eq. (1) at the knots. A similar method was used by Späth [8] to obtain an algorithm for the deficient quintic spline but with different end conditions. (He specified the second derivatives at the endpoints of the interval instead of requiring the third derivative to be zero at the endpoints as for the natural spline.)

Using eq. (1) we see at once that $S'(x_i) = y_i'$ implies $B_i = y_i'$. Then imposing the requirement that $S(x)$, $S'(x)$, and $S''(x)$ be continuous at x_{i+1}, $i = 0, 1, \ldots, n-2$, we obtain by setting $t = h_i = x_{i+1} - x_i$ in eq. (1):

$$\begin{aligned} y_{i+1} &= y_i + B_i h_i + C_i h_i^2 + D_i h_i^3 + E_i h_i^4 + F_i h_i^5 \\ B_{i+1} &= B_i + 2C_i h_i + 3D_i h_i^2 + 4E_i h_i^3 + 5F_i h_i^4 \\ C_{i+1} &= C_i + 3D_i h_i + 6E_i h_i^2 + 10F_i h_i^3. \end{aligned} \tag{7}$$

If we multiply these three equations by $10/h_i^3$, $-4/h_i^2$, $1/h_i$, respectively, and add the results, we eliminate E_i and F_i and obtain

$$D_i = 10\,\frac{y_{i+1} - y_i}{h_i^3} - \frac{4B_{i+1} + 6B_i}{h_i^2} + \frac{C_{i+1} - 3C_i}{h_i}. \tag{8}$$

We note that $D_i = S'''(x_i + 0)/6$. In order to obtain a similar expression for $S'''(x_i - 0)/6$, we replace the subscript $i+1$ by $i-1$ consistently (noting that then $h_i = x_{i+1} - x_i$ is to be replaced by $x_{i-1} - x_i = -h_{i-1}$). We obtain

$$\tfrac{1}{6}S'''(x_i - 0) = 10\,\frac{y_i - y_{i-1}}{h_{i-1}^3} - \frac{6B_i + 4B_{i-1}}{h_{i-1}^2} + \frac{3C_i - C_{i-1}}{h_{i-1}}. \tag{9}$$

Now since the third derivative is continuous at x_i, we can equate the values of $S'''(x_i + 0)/6$ and $S'''(x_i - 0)/6$ in eqs. (8) and (9) to obtain the following set of equations for the C_i:

$$-\frac{1}{h_{i-1}}C_{i-1} + \left(\frac{3}{h_{i-1}} + \frac{3}{h_i}\right)C_i - \frac{1}{h_i}C_{i+1}$$
$$= 10\left(\frac{y_{i+1} - y_i}{h_i^3} - \frac{y_i - y_{i-1}}{h_{i-1}^3}\right) - \frac{4B_{i+1} + 6B_i}{h_i^2} + \frac{6B_i + 4B_{i-1}}{h_{i-1}^2}.$$

These equations hold for $i = 1, 2, \ldots, n-1$. Two additional equations can be obtained from the conditions $S'''(x_0) = S'''(x_n) = 0$ by setting $i = 0$ in eq. (8) and by setting $i = n$ in eq. (9):

$$\frac{3}{h_0}C_0 - \frac{1}{h_0}C_1 = 10\,\frac{y_1 - y_0}{h_0^3} - \frac{4B_1 + 6B_0}{h_0^2},$$

$$-\frac{1}{h_{n-1}}C_{n-1} + \frac{3}{h_n}C_n = -10\,\frac{y_n - y_{n-1}}{h_{n-1}^3} + \frac{6B_n + 4B_{n-1}}{h_{n-1}^2}.$$

This system of $n+1$ equations is solved to obtain the C_i. If we make the substitutions

$$\begin{aligned} x &= D_i, & p &= (y_{i+1} - y_i - B_i h_i - C_i h_i^2)/h_i^3, \\ y &= E_i h_i, & q &= (B_{i+1} - B_i - 2C_i h_i)/h_i^2, \\ z &= F_i h_i^2, & r &= (C_{i+1} - C_i)/h_i, \end{aligned}$$

then for each i eqs. (7) form a system of three equations in the three unknowns x, y, z, and the system is solved by Gaussian elimination. The backward substitution yields formulas for z, y, x in the following order:

$$\begin{cases} g = q - 3p \\ z = r - 3(p + g) \\ y = g - z - z \\ x = p - y - z \\ F_i = z/h_i^2 \\ E_i = y/h_i \\ D_i = x. \end{cases}$$

3. Tests

These procedures have been tested in Algol 60 on the Telefunken TR-440 computer at the Leibniz-Rechenzentrum of the Bavarian Academy of Sciences, Munich, and in Algol W on the IBM 360/67 at the Stanford Center for Information Processing. The latter tests included timing tests of the procedures with the number of knots $N = N2 - N1 + 1$ ranging up to 1000. The time was found to be approximately proportional to the number N of knots. The time T in seconds for the execution of the procedure *QUINAT* was found to be approximately $T = .00193N$, whereas for the procedure *NATSPLINE* of Algorithm 472 [6] with $m = 3$ it was found to be $T = .0120N$, or over six times as great. For the procedure *QUINEQ* the time was approximately $T = .00064N$, whereas for the procedure *NATSPLINEEQ* of Algorithm 472 [6] with $m = 3$ it was $T = .0038N$, or nearly six times as great. For the procedure *QUINDF* the time was approximately $T = .00087N$, whereas for the procedure *QUINAT* with $2N$ knots, consecutive knots being equal in pairs, the time was $T = .00325N$, or nearly four times as great. Moreover, to compute the same results the procedure *QUINAT* requires approximately twice as much storage for the arrays used as does the procedure *QUINDF*. Note also that from the preceding formula for the time required by the procedure *QUINAT*, the time for $2N$ distinct knots would be $T = .00386N$, which can be compared with $T = .00325N$ given above for N pairs of equal knots. The reduction for the case of double knots occurs because some calculations are omitted when knots are coincident.

These timing comparisons show that it is definitely advantageous to use these special procedures for the quintic natural spline rather than the general cases given in Algorithm 472 [6] with $m = 3$.

Tests of the accuracy and correctness of the coefficients computed by the procedures *QUINAT*, *QUINEQ*, and *QUINDF* were carried out as described in Algorithm 472 [6]. Table I shows the results of a typical run using *QUINDF* for 5 nonequidistant points. The values of the function and its first derivatives were specified. The first line of each entry gives the tabulated quantities at the given value of x which is the lefthand endpoint of the subinterval; the second line of each entry gives the tabulated values at the righthand endpoint of the same subinterval. The close agreement of the quantities $S(x)$, $S'(x)$, $S''(x)/2$, and $S'''(x)/6$ shows that the quintic spline function and its derivatives satisfy the required continuity conditions. This is a good indication of the correctness of the results. Note that the

Table I. Quintic Spline Calculated by *QUINDF*
(Machine precision approximately 7 decimal digits.)

x	S(x)	S'(x)	S"(x)/2	S"'(x)/6	S""(x)/24	S^V(x)/120
-3.000000	7.000000 11.00000	2.000000 15.00000	-6.108377 7.674876	-5.722046'-06 -4.933508	2.956286 -4.189662	-0.7145951 -0.7145951
-1.000000	11.00000 26.00000	15.00000 9.999996	7.674870 -1.908880	-4.933474 16.59851	-8.157658 18.92365	5.416262 5.416262
0	26.00000 55.99942	10.00000 -27.00066	-1.908880 -5.264791	16.59848 20.03839	-9.059000 9.632320	1.246088 1.246088
3.000000	56.00000 29.00000	-27.00000 -30.00000	-5.264426 -7.754811	20.03847 -1.907349'-06	-21.28366 11.26443	6.509618 6.509618
4.000000	29.00000	-30.00000				

fourth and fifth derivatives are discontinuous. Essentially the same results were obtained by using *QUINAT* with 10 knots, in equal pairs. In addition, accuracy and timing tests were carried out for large values of N, including $N = 1000$ and 5000, and produced very satisfactory results.

REFERENCES

1. ANSELONE, P.M., AND LAURENT, P.J. A general method for the construction of interpolating and smoothing spline functions *Numer. Math* *12* (1968), 66–82.
2. CURRY, H.B., AND SCHOENBERG, I.J. On Pólya frequency functions. IV. The fundamental spline functions and their limits. *J. Anal. Math* *17* (1966), 71–107.
3. GREVILLE, T.N.E. Introduction to spline functions. In *Theory and Applications of Spline Functions*, T.N.E. Greville, Ed. Academic Press, New York, 1969, pp. 1–35.
4. GREVILLE, T.N.E. Spline functions, interpolation and numerical quadrature. In *Mathematical Methods for Digital Computers, vol. II*, A. Ralston and H.S. Wilf, Eds. Wiley, New York, 1967.
5. HERRIOT, J.G., AND REINSCH, C.H. Algol 60 procedures for the calculation of interpolating natural quintic spline functions. Tech. Rep. STAN-CS-74-402, Dep. Computer Science, Stanford Univ , Stanford, Calif., 1974.
6. HERRIOT, J G., AND REINSCH, C H. Algorithm 472, Procedures for natural spline interpolation. *Comm. ACM 16*, 12 (Dec. 1973), 763–768.
7. SCHOENBERG, I.J. On interpolation by spline functions and its minimal properties. In *On Approximation Theory. Proceedings of the Conference at Oberwolfach, 1963*, P.L. Butzer and J. Korevaar, Eds Birkhaüser Verlag, Basel, Switzerland, 1964, pp 109–129.
8. SPÄTH, H. Algorithm 42, Interpolation by certain quintic splines. *Computer J. 12* (1969), 292–293.

(*Please see algorithm on next page*)

ALGORITHM

[Only that portion of the listing which gives the introductory comments explaining the algorithm is printed here. The complete listing is available from the ACM Distribution Service (see inside back cover for order form), or may be found in "Collected Algorithms from ACM."]

```
PROCEDURE QUINAT(INTEGER VALUE N1,N2; REAL ARRAY X,Y,B,C,D,E,F(*));
COMMENT  QUINAT COMPUTES THE COEFFICIENTS OF A QUINTIC NATURAL SPLINE
   S(X) INTERPOLATING THE ORDINATES Y(I) AT POINTS X(I), I = N1
   THROUGH N2.  FOR XX IN (X(I),X(I+1)) THE VALUE OF THE SPLINE
   FUNCTION S(XX) IS GIVEN BY THE FIFTH DEGREE POLYNOMIAL:
   S(XX) = ((((F(I)*T+E(I))*T+D(I))*T+C(I))*T+B(I))*T+Y(I)
   WITH T = XX - X(I).
   INPUT:
     N1,N2  SUBSCRIPT OF FIRST AND LAST DATA POINT RESPECTIVELY,
            IT IS REQUIRED THAT N2 > N1 + 1,
     X,Y(N1::N2)  ARRAYS WITH X(I) AS ABSCISSA AND Y(I) AS ORDINATE
            OF THE I-TH DATA POINT.  THE ELEMENTS OF THE ARRAY X
            MUST BE STRICTLY MONOTONE INCREASING (BUT SEE BELOW FOR
            EXCEPTIONS TO THIS).
   OUTPUT:
     B,C,D,E,F(N1::N2)  ARRAYS COLLECTING THE COEFFICIENTS OF THE
            QUINTIC NATURAL SPLINE S(XX) AS DESCRIBED ABOVE.
            SPECIFICALLY  B(I) = S'(X(I)),  C(I) = S"(X(I))/2,
            D(I) = S"'(X(I))/6,  E(I) = S""(X(I))/24,
            F(I) = S""'(X(I)+0)/120.  F(N2) IS NEITHER USED OR
            ALTERED.  THE ARRAYS B,C,D,E,F MUST ALWAYS BE DISTINCT.
   OPTIONS:
     1.  THE REQUIREMENT THAT THE ELEMENTS OF THE ARRAY X BE
         STRICTLY MONOTONE INCREASING CAN BE RELAXED TO ALLOW TWO
         OR THREE CONSECUTIVE ABSCISSAS TO BE EQUAL AND THEN
         SPECIFYING VALUES OF THE FIRST AND SECOND DERIVATIVES OF
         THE SPLINE FUNCTION AT SOME OF THE INTERPOLATING POINTS.
         SPECIFICALLY
         IF X(J) = X(J+1) THEN S(X(J)) = Y(J) AND S'(X(J)) = Y(J+1),
         IF X(J) = X(J+1) = X(J+2) THEN IN ADDITION S"(X(J)) =Y(J+2).
         NOTE THAT S""(X) IS DISCONTINUOUS AT A DOUBLE KNOT AND IN
         ADDITION S"'(X) IS DISCONTINUOUS AT A TRIPLE KNOT.  AT A
         DOUBLE KNOT, X(J) = X(J+1), THE OUTPUT COEFFICIENTS HAVE THE
         FOLLOWING VALUES:
           B(J) = S'(X(J))       = B(J+1)
           C(J) = S"(X(J))/2     = C(J+1)
           D(J) = S"'(X(J))/6    = D(J+1)
           E(J) = S""(X(J)-0)/24    E(J+1) = S""(X(J)+0)/24
           F(J) = S""'(X(J)-0)/120  F(J+1) = S""'(X(J)+0)/120
         THE REPRESENTATION OF S(XX) REMAINS VALID IN ALL INTERVALS
         PROVIDED THE REDEFINITION Y(J+1) := Y(J) IS MADE
         IMMEDIATELY AFTER THE CALL OF THE PROCEDURE QUINAT.  AT A
         TRIPLE KNOT, X(J) = X(J+1) = X(J+2), THE OUTPUT COEFFICIENTS
         HAVE THE FOLLOWING VALUES:
           B(J) = S'(X(J))        = B(J+1) = B(J+2)
           C(J) = S"(X(J))/2      = C(J+1) = C(J+2)
           D(J) = S"'(X(J)-0)/6   D(J+1) = 0  D(J+2) = S"'(X(J)+0)/6
           E(J) = S""(X(J)-0)/24  E(J+1) = 0  E(J+2) = S""(X(J)+0)/24
           F(J) = S""'(X(J)-0)/120  F(J+1)=0  F(J+2)=S""'(X(J)+0)/120
         THE REPRESENTATION OF S(XX) REMAINS VALID IN ALL INTERVALS
         PROVIDED THE REDEFINITION Y(J+2) := Y(J+1) := Y(J) IS MADE
         IMMEDIATELY AFTER THE CALL OF THE PROCEDURE QUINAT.
       2. THE ARRAY X MAY BE MONOTONE DECREASING INSTEAD OF
          INCREASING;
IF N2 > N1 + 1 THEN
BEGIN
```