Reprinted from
Henry L. Garabedian (editor)
Approximation of Functions
Elsevier Publishing Company,
Amsterdam 1965
Printed in Northern Ireland

PIECEWISE POLYNOMIAL INTERPOLATION AND APPROXIMATION[1]

GARRETT BIRKHOFF
Harvard University, Cambridge, Massachusetts

and

CARL R. DE BOOR
General Motors Research Laboratories, Warren, Michigan

A. Introduction

1. **Motivation.** Practical interest in schemes of interpolation and approximation has been greatly stimulated by the development of high-speed digital computing machines having a large storage ("memory"). Moreover, the availability of such machines has also heightened interest in the question of representing more or less arbitrary curves and surfaces by relatively simple formulas. In particular, the automobile industry would like to represent car body surfaces by formulas which could be handled by such machines.

We have been working on this problem for several years, in collaboration with the mathematics staff at the General Motors Research Laboratories, with the primary aim of developing simple, economical, accurate and flexible procedures. From the beginning, it has been our conviction that *piecewise polynomial* functions were the most suitable, and our experience indicates that they are basically satisfactory. We wish here to explain *why* we believe that they are well suited for treating *general* problems of interpolation and approximation on high-speed computing machines.

The theoretical literature on piecewise polynomial functions constitutes a small fraction of the existing mathematical literature on interpolation and approximation theory. The bulk of this literature has been concerned with the fitting of functions of *one* real or complex variable by *analytic* functions. Interpolation and approximation by *polynomial* and *rational* functions offer a wealth of alternative possibilities [**10**, Chapter VIII; **7**, **8**, **16**], and many theorems have been proved about the *convergence* of such schemes as the degree tends to infinity. However, as is well known, simple polynomial interpolation fails to converge as the mesh-length tends to zero, even for some very smooth analytic functions (e.g., $1/(1 + x^2)$ on $[-5, 5]$),[2] and uniform meshes.

For *periodic* functions tabulated on a *uniform mesh* (i.e., one with constant

[1] Work partly supported by the Office of Naval Research.

[2] J. F. Steffensen, *Interpolation*, 2nd ed., Chelsea, New York, 1950, p. 35. The counterexample is due to C. Runge, Z. Math. Phys. **46** (1901), 224–243.

mesh-spacing h), a reliable scheme of accurate analytic interpolation is provided by truncated Fourier series (trigonometric polynomials). This scheme was extended from truncated Fourier series to truncated Fourier integrals, by E. T. Whittaker, in his classic study of "cardinal functions" [**18**; **8**, p. 330]. These define an interpolation scheme for functions tabulated on a uniform mesh with mesh-points $x_n = nh$ $(n = 0, \pm 1, \pm 2, \cdots)$, which is ideal for many theoretical purposes. For any continuous, square-integrable function f in the Hilbert space $L_2(-\infty, +\infty)$ it reproduces exactly the component f_h from wave-numbers $q \in [-\pi/h, \pi/h]$, but replaces each Fourier component with wave-number $q \notin [-\pi/h, \pi/h]$ by a "cotabular" function with the same coefficient but different wave number q' satisfying $q' \equiv q \pmod{2\pi/h}$ and $|q'| \leqq \pi/h$. Hence it rotates the orthogonal component $f - f_h$ through 90° in the Hilbert space. It is *convergent* in the mean-square, and the error is $O(h^r)$ if the rth derivative $f^r(x) \in L_2(-\infty, +\infty)$.

However these schemes are inflexible (the mesh-spacing must be constant); they are incompatible with most boundary conditions; and we shall see (§11) that they are sensitive to roundoff errors.

Much less is known about the more difficult problems of interpolation and approximation to functions of two or more variables. The Weierstrass Approximation Theorem assures us that *any* continuous function can be *approximated* arbitrarily closely on *any* compact set by polynomials of sufficiently high degree [**4**, §6.6]. Approximations can be computed systematically, as convolutions with Bernstein polynomials; moreover one can also match finite sets of derivatives arbitrarily closely [**4**, §6.3]. From the standpoint of *existence* theory, this leaves little to be desired. Moreover, because high-speed computing machines have arithmetic units especially designed to perform rational operations, polynomials and rational functions are well adapted to them.

However, even for functions of one variable, the use of Bernstein polynomials to *compute* accurate polynomial approximations is *uneconomical* [**4**, p. 116]. In practice, Newton's and Lagrange's interpolation formulas are far more widely used, even though they may diverge if pushed too far.

2. **Piecewise polynomial functions.** To represent smooth curves and surfaces economically, with the help of high-speed digital computers, we recommend the use of *piecewise polynomial* functions.

Indeed, we will be much more specific. Although piecewise quintic polynomials have proved useful on occasion and, for some applications, approximation by piecewise quadratic or even linear functions is most suitable, in general we recommend *piecewise cubic* polynomials for fitting smooth curves, and *piecewise bicubic* polynomials for fitting smooth surfaces, as good bets to be tried first, in the absence of special reasons for trying something else.

Piecewise cubic polynomial functions of one variable, with continuous slope and curvature, have long been used by draftsmen and engineers. For practical design work, they have used *mechanical splines*: thin beams carrying loads w_i

concentrated at points x_i, according to the classical Euler-Bernoulli theory.[3] Such mechanical splines (of small "stiffness") have been used as analog computers to fair curves through given sets of points.

It was probably known to Euler that the "strain energy" minimized by such splines is proportional to $\int y''^2\,dx$ in the small-deflection (linearized) approximation. Moreover, the shape of "non-linear splines" (or elastica) minimizing $\int \kappa^2\,ds$, the integral of the squared curvature with respect to arc-length, has also been worked out **[9**, §262].

The use of mechanical splines to interpolate smooth nearly horizontal curves through a given set of points is similar to that of computing the deflection of a *thin beam* of given stiffness k. In both cases, the third derivative undergoes a jump of $w_i/k = \Delta y_i'''$ at the ith joint x_i, and the deflection $y(x)$ is given (approximately) by a cubic polynomial between successive "joints" x_i. The difference is, that in the problem of spline *interpolation* one is *given* the $y_i = y(x_i)$, whereas in the problem of the loaded beam one is given the w_i (or $\Delta y_i'''$). Moreover by *clamping* the spline at any joint, one can control the slope there.

By using clamped splines, one can represent very accurately horizontal plane sections of ship hulls.[4] Typically, ship hulls have long straight midsections, onto which a smooth, pointed bow and stern are appended. The advantages of using splines or other piecewise polynomial functions to represent such profiles seems indeed fairly obvious, if one considers the principle of analytic continuation.

It can hardly be said that this idea is either very deep or very novel. The use of generalized splines, and other piecewise polynomial functions of higher degree, to approximate smooth functions of one variable, was considered earlier very carefully by I. J. Schoenberg in an important paper **[13]** where deep results were obtained for the case of a uniform mesh on the infinite line. A thorough study of the literature would probably reveal many other relevant papers.

What is important is that spline interpolation *converges* rapidly on a wide variety of meshes, that it is insensitive to roundoff, and that it is easy to perform on high-speed computers. The demonstration of these facts seems to be new. What we consider most original about our work is the development of practical schemes of *surface-fitting*, applicable to wide classes of smooth surfaces.

B. Piecewise Polynomial Interpolation

3. **Spline functions.** To introduce the subject technically, we will adopt the approach of Schoenberg **[13]**. We define a *spline function of degree k*, with *joints* at points $x_0 < x_1 < \cdots < x_n$, as a function $f(x)$ of one real variable x, which is

[3] See for example J. L. Synge and B. A. Griffith, *Principles of mechanics*, McGraw-Hill, New York, 1959, §3.3.

[4] See F. Theilheimer and W. Starkweather, MTAC **15** (1961), 338–355. It was in connection with this application that the senior author first became attracted to spline functions, around 1955.

of class C^{k-1}, and is equal on each interval $[x_{i-1}, x_i]$ to a polynomial of degree k.

Spline functions of *odd* degree $k = 2m - 1$ have the basic variational property of minimizing the integral $\int [f^{(m)}(x)]^2\,dx$ for given $y_i = f(x_i)$; this is immediate if one integrates by parts. Thus, cubic spline functions ($m = 2$) appear as the logical next step after piecewise linear ("polygon") functions of x (the case $m = 1$), as a scheme of piecewise polynomial interpolation. For given y_i and endslopes y_0', y_n', they are easily calculated as follows.

The condition that $f(x) \in C^2$, for $f(x)$ cubic in the intervals $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$, is equivalent to the following linear equation:

$$(3.1)\quad \Delta x_i y_{i-1}' + 2(\Delta x_{i-1} + \Delta x_i) y_i' + \Delta x_{i-1} y_{i+1}' = 3[(\Delta x_i \Delta y_{i-1}/\Delta x_{i-1}) + (\Delta x_{i-1} \Delta y_i/\Delta x_i)],$$

where $\Delta x_i = x_{i+1} - x_i$ and $\Delta y_i = f(x_{i+1}) - f(x_i)$. The resulting system of $n - 1$ equations is not only linearly independent; it is tridiagonal and *diagonally dominant*; hence it can be very stably solved for interior y_i' if y_0', y_n' are given. Having solved for y_i and y_i' $[i = 0, 1, \cdots, n]$, one easily computes $f(x)$ in each interval $[x_{i-1}, x_i]$ by Hermite interpolation.

The preceding method can be adapted to cover also the "free endpoint" conditions $y_0'' = y_n'' = 0$. In this case, one must supplement (3.1) by

$$(3.1')\qquad 2y_0' + y_1' = 3\Delta y_0/\Delta x_0, \qquad y_{n-1}' + 2y_n' = 3\Delta y_{n-1}/\Delta x_{n-1}.$$

Unfortunately, a valuable property is lost when one passes from $m = 1$ to $m = 2$. Whereas broken line interpolation is *local*, in the sense that the value of the interpolating function at a given point depends only on the f_i at a *fixed finite set* of neighboring points, this is not true of cubic spline interpolation. The values of $f(x)$ in any interval $[x_{i-1}, x_i]$ depend on *all* $f(x_j)$, without exception.

Local piecewise polynomial interpolation. In this respect, Bessel's method of *local cubic* interpolation has an advantage over spline interpolation. In local cubic interpolation, the numbers $y_1', y_2', \cdots, y_{n-1}'$ are calculated from

$$(3.2)\qquad (\Delta x_{i-1} + \Delta x_i) y_i' = \left[\Delta x_i \frac{\Delta y_{i-1}}{\Delta x_{i-1}} + \Delta x_{i-1} \frac{\Delta y_i}{\Delta x_i}\right];$$

Hermite interpolation is then again used to compute $f(x)$ in each interval. This gives a piecewise cubic polynomial of class C^1 only whose value at any point $x \in [x_{i-1}, x_i]$ depends just on the eight numbers x_j, y_j $(j = i - 2, i - 1, i, i + 1)$.

One can achieve continuity of the second derivative by increasing the degree of the polynomial pieces (and the number of points used). Jenkin's interpolation formula **[13]**, for instance, uses quartic polynomials, to obtain an interpolating function of class C^2 whose value in most intervals $[x_{i-1}, x_i]$ depends on the six numbers $y_{i-3}, \cdots, y_{i+1}$. For intervals near the endpoints x_0, x_n, it must be altered.

More generally, (local) interpolation formulas can be easily constructed **[13, 11]** for any positive integer m, which yield functions $f \in C^m$ which are piecewise polynomial of degree $k \leq m + 1$, with "joints" at the x_i. However, it is our impression that such formulas are more cumbersome to use than spline formulas of equal accuracy.

4. **Convergence to curves.** Let π: $0 = x_0 < x_1 < \cdots < x_n = 1$ be a partition of the unit interval, and, for a given function $f(x)$, let $s(x) = s(x, \pi, f)$ denote the cubic spline which satisfies $s(x_i) = f(x_i)$, $i = 0, 1, \cdots, n$, and $s'(0) = f'(0)$, $s'(1) = f'(1)$. What happens to the error $s(x) - f(x)$ as $|\pi| = \max \Delta x_i$ shrinks to zero?

A first answer to this question was given by Ahlberg and Nilson **[1]**, where it was stated that if $f(x) \in C^2[0, 1]$, and if the mesh becomes eventually uniform as $|\pi| \to 0$, then $s(x)$ and its first two derivatives converge uniformly to $f(x)$ and its first two derivatives. We have extended this result in **[2]**, by showing that if $f(x) \in C^4[0, 1]$, and if the mesh-ratio $M_\pi = \max (\Delta x_i / \Delta x_j)$, the maximum ratio of mesh-lengths, is bounded, then $s^{(j)}(x)$ converges uniformly to $f^{(j)}(x)$, $j = 0, \cdots, 3$, as $|\pi| \to 0$. In fact,

$$|s^{(j)}(x) - f^{(j)}(x)| \leqq K \cdot |\pi|^{4-j}. \tag{4.1}$$

The uniform convergence of $s^{(3)}(x)$ to $f^{(3)}(x)$ was also proved, assuming only that $f^{(3)}(x)$ was absolutely continuous.

The technique used in **[2]** was based on a study of the *cardinal functions* $C_i(x)$ associated with spline interpolation at the joints x_i of a given partition π. These are the spline functions with joints x_i defined by

$$C_i(x_j) = \delta_{ij}, \qquad C_i'(x_0) = C_i'(x_n) = 0, \qquad i = 1, \cdots, n - 1, \qquad j = 0, \cdots, n. \tag{4.2}$$

For bounded M_π, these are uniformly bounded and integrable—in fact they die away exponentially by a factor appreciably less than one in *each successive interval*.

When $f \in C^4[0, 1]$, formula (4.1) implies that the approximation to $f'(x)$ by $s'(x)$ is $O(|\pi|^3)$. A stronger statement can be made in case $x = x_i$ for some i, and π is uniform. In this case equation (3.1) reduces to

$$\frac{h}{3}(s'(x_{i-1}) + 4s'(x_i) + s'(x_{i+1})) = s(x_{i+1}) - s(x_{i-1}), \tag{4.3}$$

where $h = |\pi|$. One recognizes (4.3) as Simpson's Rule, and concludes from this (or computes directly) that

$$s'(x_{i-1}) + 4s'(x_i) + s'(x_{i+1}) = f'(x_{i-1}) + 4f'(x_i) + f'(x_{i+1}) + O(|\pi|^4), \tag{4.4}$$

in case $f \in C^5[0, 1]$ (since $s(x_j) = f(x_j)$). Hence, since $s'(0) = f'(0)$, $s'(1) = f'(1)$, and the eigenvalues of the tridiagonal matrix with general row $\{1, 4, 1\}$ are all greater than 1, it follows that

$$s'(x_i) = f'(x_i) + O(|\pi|^4).$$

It is clear that this conclusion remains true if the condition that π is uniform is replaced by the condition that π becomes asymptotically uniform as $|\pi| \to 0$. Unfortunately, the analogous result does not hold for s'': even at the joints, $s''(x) = f''(x) + O(|\pi|^2)$.

5. **Convergence and continuity.** We will now introduce some general concepts, which characterize valuable features of the (cubic) spline interpolation scheme defined in §3. An interpolation scheme J will be called *algebraically well-defined* for a real closed domain D and class Π of meshes π on D when, for any function[5] $f \in C^\infty(D)$, any $\pi \in \Pi$, $J_\pi[f]$ exists and is uniquely determined. Here, it is understood that *derivatives* of f may be admitted as values, as limiting cases (Hermite interpolation). Clearly, Lagrange, trigonometric, and (cubic) spline interpolation are all algebraically well-defined, linear interpolation schemes.

For an interpolation scheme to be a good scheme of approximation, a small maximum mesh-length $|\pi|$ must result in a small interpolation error $(J_\pi f)(x) - f(x)$. This leads to the question of whether or not a given interpolation scheme J is *convergent* on a set $S \subset C(D)$ of continuous functions on D, in the sense that

$$\max_{x \in D} |(J_\pi f)(x) - f(x)| \to 0 \quad \text{for} \quad \pi \in \Pi, \quad \text{as} \quad |\pi| \to 0. \tag{5.1}$$

Usually, one takes for S the set $C^k(D)$, for some $k \geqq 1$, and can often derive statements about the order ν of convergence as well, where ν is defined as the largest integer for which

$$\max_{x \in D} |(J_\pi f)(x) - f(x)| = O(|\pi|^\nu), \qquad \pi \in \Pi, \qquad f \in C^k(D). \tag{5.2}$$

Thus equal-spaced trigonometric interpolation is convergent on $C^k[a, b]$ for $k \geqq 1$, the interpolation error being $O(|\pi|^k \ln |\pi|)$ **[7]**. We have just seen that spline interpolation on $[a, b]$ for given $s'(a) = f'(a)$, $s'(b) = f'(b)$ *is* convergent on $C^4[a, b]$, with order of convergence equal to 4, provided Π consists of partitions with uniformly bounded mesh-ratios M_π. By contrast, as remarked earlier, polynomial or Lagrange interpolation (on uniform meshes) is not convergent even on $C^\infty(D)$ for all analytic functions! Also, not all schemes of spline interpolation are convergent. For example, if the condition $s'(b) = f'(b)$ is replaced by $s''(a) = f''(a)$ or by $s'((a + b)/2) = f'((a + b)/2)$, one would get algebraically well-defined schemes which are not convergent even on $C^\infty[a, b]$.

[5] Note that any set of values and derivatives can be assumed by some function $f \in C^\infty(D)$.

Theoretical results on the orders of convergence of interpolation schemes applied to a sufficiently smooth function f lose some of their practical interest because of roundoff. Although a smooth function f is to be interpolated, random roundoff errors will result in a non-smooth interpolating function g, the maximum difference between f and g being of the order of magnitude of the roundoff. Hence, it is essential for practical purposes that a small $|\pi|$ *and* a small roundoff error ε imply a small interpolation error. This demands *continuity* of the interpolation scheme in the sense that

$$\begin{aligned} &\max_{x\in D} |f_\varepsilon(x) - f(x)| \to 0, \quad \text{and} \quad |\pi_n| \to 0, \qquad \pi_n \in \prod, \\ &\text{imply} \quad \max_{x\in D} |(J_{\pi_n} f_\varepsilon)(x) - f(x)| \to 0, \qquad \text{as} \quad \varepsilon \to 0. \end{aligned} \tag{5.3}$$

If J is a scheme of linear interpolation by continuous functions defined on a dense subset of $C(D)$, then J_π is a linear operator from the Banach space $C(D)$ (with respect to the uniform norm $\|f\| = \max_{x\in D} |f(x)|$) into itself. In this case, J is continuous if and only if J is convergent on $C(D)$. This is a corollary of Banach's "uniform boundedness principle": J is convergent on $C(D)$ if and only if J is convergent on a dense subset of $C(D)$ and the norms of J_{π_n} are uniformly bounded as $|\pi_n| \xrightarrow[n\to\infty]{} 0$. It is analogous to the Lax-Richtmyer theorem for difference approximations to differential operators.

The main result of this section is the fact that the (cubic) spline interpolation scheme of §3 is indeed continuous. For, $(J_\pi f)(x) = \sum_i f(x_i)C_i(x)$, where $C_i(x)$ is the cardinal function of spline interpolation corresponding to the partition π (cf. §3). But, as was shown in **[2]**, $\sum_i |C_i(x)| \leqq K$ for some fixed K depending only on the maximum mesh ratio M_π. Hence, if π is a set of partitions with uniformly bounded mesh-ratio, then, for all $\pi \in \Pi$,

$$|(J_\pi f)(x)| \leqq \sum_i |f(x_i)|\,|C_i(x)| \leqq \|f\| \cdot \sum_i |C_i(x)| \leqq K \cdot \|f\|,$$

so that the operators J_π are indeed uniformly bounded, while $J_{\pi_n} f$ converges to f for all f in the dense subset $C^4[a, b]$ of $C[a, b]$.

This situation is in notable contrast with approximation by cardinal and trigonometric interpolation. That the latter is not continuous is well known.[6] As to cardinal interpolation, consider the cardinal function for cardinal interpolation with mesh-length h, $C_i(x - ih) = [\sin(\pi x/h)]/(\pi x/h)$. Since this is square-summable over i, we see that the cumulative effect of *independent* random roundoff errors of bounded size remains bounded as $h\downarrow 0$. Since it is not absolutely summable, the *cumulative* error due to *systematic* roundoff errors with alternating sign and fixed order of magnitude would be unbounded, if one lets h tend to zero.

In general, we surmise that all schemes of interpolation by *analytic* functions will tend to be sensitive to roundoff errors, and any other local irregularities. This is because of the principle of analytic continuation, which makes the behavior of an analytic interpolating function in any neighborhood, however small, determine exactly its behavior everywhere.

[6] Cf. A. Zygmund, *Trigonometric series*, 2nd ed., Cambridge Univ. Press, London, 1959, Vol. 2.

6. **Non-linear spline interpolation.** Linearized interpolation schemes have a basic shortcoming: they are *not intrinsic* geometrically because they are not invariant under rigid rotation. Physically it seems more natural to replace linearized spline curves by *non-linear* splines (or "elastica"), well known among elasticians **[9**, §262], and this idea has been carefully considered by various people, including ourselves. Indeed, at least two computational schemes of non-linear spline interpolation have been proposed in the literature.

One scheme, proposed by A. H. Fowler and C. W. Wilson at Oak Ridge **[27]**, goes in principle as follows. Choose coordinates for each segment $\widehat{P_{i-1}P_i}$ so that the x-axis is parallel to the straight line $\overline{P_{i-1}P_i}$, and approximate $\widehat{P_{i-1}P_i}$ by a cubic polynomial (linearized spline segment) in these coordinates. Then require continuity of slope *and* curvature at all interior mesh-points. This requires the iterative solution of a non-linear system of equations. A simplified discussion of this scheme, written by the General Motors Research Laboratory staff, is available as a research report **[21]**.

The second scheme, proposed by D. H. MacLaren **[28]** at Boeing in 1959, sacrifices continuity of curvature, but has the advantage of being linear. In each interval, he approximates the curvature κ by $\bar\kappa = y''/[1 + (\Delta y/\Delta x)]^{3/2}$, and minimizes $\int \bar\kappa^2\, ds$.

Mechanical splines. The preceding schemes are intended to approximate true mechanical splines, or "elastica" constrained to pass through a fixed sequence of points by pure *shear forces*. (Only one of the family of curves graphed by Love **[9**, p. 404] represents such a mechanical spline.) These curves extremalize the integral $\int \kappa^2\, ds$, which is proportional to the elastic strain energy[7]; they satisfy $\delta \int \kappa^2\, ds = 0$. This equilibrium is *stable* if and only if the extremum is a local minimum of $\int \kappa^2\, ds$.

Curiously, an absolute minimum to $\delta \int \kappa^2\, ds$ does not exist except in the trivial case of a straight line; this is because one can construct large loops joining given endpoints with given endslopes, of length $2\pi r$ and curvature $\kappa = O(1/r)$, for arbitrarily large r—hence with $\int \kappa^2\, ds$ less than any preassigned positive number. Related to this, is the absence of an existence and uniqueness theory for non-linear spline curves having given endpoints, endslopes, and passing through a given sequence of internal joints.

After looking carefully into the relevant equations, one realizes that the schemes of **[27]** and **[28]** do not approximate mechanical splines more closely than

[7] In general, an elastica extremalizes $\int (\kappa^2 - \lambda_i)\, ds$, where λ is a parameter, constant in each segment, and depending on tension and bending moment at the joints; see **[21]**.

other curves—nor does it seem particularly desirable to have them do so. For example, they approximate equally well to Hermite interpolation by segments of *Euler's spirals*,[8] joined together with continuous curvature. And Euler's spirals seem as natural a class of curves as the curves defined by "elastica" under pure shear forces. (The latter satisfy $2d^2\kappa/ds^2 + \kappa^3 = 0$.)

C. Interpolation: Functions of Two Variables

7. **Smooth surface interpolation.** The preceding discussion still fails to touch the basic problem of convergent interpolation for functions of two variables. However, it suggests an important first step towards solving the problem of fitting smooth surfaces.[9]

Namely, let offsets $u_{ij} = u(x_i, y_j)$ be given on a rectangular grid of points, the vertices of a *rectangular network* of lines including all the sides (boundary lines) of a *rectangular polygon* R. On the boundary vertices, let also the normal derivative (slopes) $\partial u/\partial n$ be given.[10] Then it is clear that exactly one network of linearized (cubic) *spline curves* can be passed through the given points, subject to the given boundary conditions.

The preceding construction raises the question: how can one interpolate *surface elements* in the rectangular pieces R_{ij}: $[x_{i-1}, x_i] \times [y_{i-1}, y_j]$, so as to obtain a *smooth surface*. In principle, one very simple answer to this question is the following.

At each vertex (x_i, y_j), the incident spline curves

$$u = u(x_i, y) = f_i(y) \quad \text{and} \quad u = u(x, y_j) = g_j(x) \tag{7.1}$$

have well-defined slopes $f_i'(y)$ and $g_j'(x)$. Why not *interpolate linearly* in $\partial u/\partial n$ along each edge of R_{ij}, thus achieving a joint of class C^1? To solve the problem of smooth surface interpolation in R for the given data, therefore, it suffices to find a 12-*parameter family* of functions determined by the values of u, $\partial u/\partial x$ and $\partial u/\partial y$ on the corners of each R_{ij}, interpolating to u by Hermite interpolation and to $\partial u/\partial n$ by linear interpolation along the edge.

This defines a classic boundary value problem in the theory of elasticity, associated with the *biharmonic* equation $\nabla^4 u = 0$; its solution minimizes

$$\iint (\nabla^2 u)^2 \, dx \, dy.$$

---

[8] These are curves defined by the relation $d^2\kappa/ds^2 = 0$ (whence $\kappa = a_i s + b_i$ on $\widehat{P_{i-1}P_i}$); see Am. Math. Monthly **25** (1918), 276–282.

[9] We ignore the scheme of bilinear interpolation, since it gives surfaces with edges except in trivial cases.

[10] At projecting corners, both normal derivatives; at reentrant corners, neither derivative. As an alternative, one can use the "free endpoint" condition $\partial^2 u/\partial n^2 = 0$.

Since this is an exact two-dimensional analog of the spline problem, its solution constitutes a natural method of surface fitting, proposed in [3]. There, it was noted that eight linearly independent *polynomial* solutions are available: $1, x, y, x^2, xy, y^2, x^3, y^3$.

However, correspondence with Prof. E. Sternberg of Brown University made it apparent that the other four would be very hard to compute. Therefore Dr. Garabedian and one of us concocted somewhat arbitrarily four additional *piecewise polynomial* functions (F_6 F_9, F_{11}, F_{12} in the notation of [3]) which seemed adequate for the purpose in hand. These functions gave a surface of class C^1 which satisfied the stated boundary conditions. Though it was not of class C^2, it seemed clear that *no* solution of the problem which had piecewise linear $\partial u/\partial n$ along the interfaces $x = x_i$ and $y = y_j$ could have a continuous cross-derivative $u_{xy} = \partial^2 u/\partial x\, \partial y$. Hence, the method of surface interpolation proposed in [3] seems in some sense nearly "best possible" within the framework of a 12-parameter family of surface elements.

8. **Bicubic spline interpolation.** To get a solution in C^2, one must abandon the use of (piecewise) linear interpolation for getting $\partial u/\partial n$ along the edges of R_{ij} from the values of u_x, u_y at the vertices obtained from the network of splines with joints at these vertices. Instead, one must use *spline* interpolation in u_x, u_y (that is, in $\partial u/\partial n$) as well. This requires using in each R_{ij} the family of all *bicubic polynomials*

$$u(x, y) = a_{00} + a_{10}x + a_{01}y + \cdots + a_{33}x^3y^3 = \sum_{i=1}^{3}\sum_{j=1}^{3} a_{ij}x^iy^j. \tag{8.1}$$

Algebraically, the bicubic polynomials (8.1) are much simpler and more convenient to use than the elaborate functions F_6, F_9, F_{11}, F_{12} mentioned in §7. In each R_{ij}, there is one and only one bicubic polynomial (8.1) which takes on specified values of u, u_x, u_y, and u_{xy} at the four corners. Moreover, if values of these quantities are specified at all mesh-points of a rectangular grid in a rectangular polygon, and Hermite interpolation is used to fill in each rectangular element, then the function $u(x, y)$ obtained by splicing these elements together is automatically of class C^1.

If the u_{ij} are given, together with the values of $\partial u/\partial n$ at all boundary mesh-points, then u_x and u_y can be computed as in §4 to give the rectangular network of spline curves mentioned in §5. For these values of u_x, u_y, the resulting piecewise bicubic polynomial function will be not only of class C^1, but will have continuous u_{xx}, u_{yy} along mesh-lines. By specifying u_{xy} at mesh-points from approximate formulas, such as

$$u_{xy} \doteq \frac{u_{i+1,j+1} - u_{i+1,j-1} + u_{i-1,j-1} - u_{i-1,j+1}}{(x_{i+1} - x_{i-1})(y_{j+1} - y_{j-1})},$$

one can hope to get quite smooth interpolating functions.

However, to get interpolating (piecewise bicubic) functions of class C^2, more ingenuity is required: one must use spline interpolation in u_x and u_y (that is, in $\partial u/\partial n$), as well as in u.

The fact that simultaneous spline interpolation in the $(u_x)_{ij}$ and $(u_y)_{ij}$ gives *consistent* values of the $(u_{xy})_{ij}$, and is compatible with (8.1), can be most easily proved by using the notion of *tensor products* of functions, as follows. (See [5] for the original proof that bicubic spline interpolation is algebraically well-defined and more details.)

Given the joints $x_0 < x_1 < \cdots < x_m$, let $C_i(x)$ $(i = 0, \cdots, m+2)$ be cubic spline functions with joints at the x_i such that

$$f(x) = \sum_{i=0}^{m} z_i C_i(x) + z_0' C_{m+1}(x) + z_m' C_{m+2}(x) \tag{8.2}$$

satisfies $f(x_i) = z_i, f'(x_0) = z_0', f'(x_m) = z_m'$. Let $D_j(y)$ be a corresponding basis for the cubic spline interpolation problem of §3, for given $g(y_j)$, $(j = 0, \cdots, n)$, $g'(y_0)$, $g'(y_n)$. Consider the functions

$$u(x, y) = \sum_{i=0}^{m+2} \sum_{j=0}^{n+2} a_{ij} C_i(x) D_j(y). \tag{8.3}$$

Each $u(x, y)$ is of class C^2 on $x_0 \leqq x \leqq x_m$; $y_0 \leqq y \leqq y_n$. Moreover

$$\begin{aligned} a_{ij} &= u_{ij}, & i &= 0, \cdots, m; & j &= 0, \cdots, n; \\ a_{ik} &= \frac{\partial}{\partial y} u(x_i, y_k), & i &= 0, \cdots, m; & k &= 0, n; \\ a_{kj} &= \frac{\partial}{\partial x} u(x_k, y_j), & k &= 0, m; & j &= 0, \cdots, n; \\ a_{ij} &= \frac{\partial^2}{\partial x\, \partial y} u(x_i, y_j), & i &= 0, m; & j &= 0, n. \end{aligned} \tag{8.4}$$

Hence, given the values at all mesh-points and the normal derivatives at the boundary mesh-points as in §7, and in addition the cross-derivative at the four corners, this interpolating function of class C^2 is determined *uniquely*.

Along each mesh-line, the surface reduces to a cubic spline, as in §7. But along each mesh-line, $\partial u/\partial n$ is a cubic spline also, as desired. In each rectangle, u is a bicubic polynomial (8.1). Its sixteen coefficients can be computed, once u, u_x, u_y, *and* u_{xy} are known at the four corners of R_{ij}. We can compute the u_x, u_y as in §3, from (3.1)–(3.1′). As $u(x, y)$ is of class C^2, we can then compute u_{xy} by spline interpolation of the u_x values along mesh-lines with constant x, or of the u_y values along mesh-lines with constant y. For example, we can solve

$$\begin{aligned} \Delta y_{j-1} s_{i,j+1} + 2(\Delta y_{j-1} + \Delta y_j) s_{ij} + \Delta y_j s_{i,j-1} \\ = 3[\Delta y_{j-1} \Delta_j p_{ij}/\Delta y_j + \Delta y_j \Delta_j p_{i,j-1}/\Delta y_{j-1}], \end{aligned} \tag{8.5}$$

as in [5. Equation (14)] for $s = u_{xy}$ from the values of $p = u_x$ at mesh-points.

It is of interest to generalize the preceding construction to *rectangular polygons*; see Appendix A.

Variational property. Bicubic spline functions can also be characterized by a variational property. Namely, the bicubic spline function satisfying (8.4) minimizes

$$\iint_R \left(\frac{\partial^4 u}{\partial x^2\, \partial y^2}\right)^2 dx\, dy + \int_E \left(\frac{\partial^2 u}{\partial s^2}\right)^2 ds, \tag{8.6}$$

subject to the constraints (8.4). Here E is the edge of R, and $\partial/\partial s$ signifies the tangential derivative.

If only the u_{ij} are prescribed, the minimum of (8.6) defines that spline function with the joints specified, having "free edges" with $\partial^2 u/\partial n^2 = 0$ on E. (In the case $I = J = 1$ of one rectangle, the resulting function is then bilinear!)

9. **Convergence to surfaces.** The results of §4 apply to the errors in smooth surface interpolation and bicubic spline interpolation as well.

Let $f(x, y) \in C^5$ on $0 \leqq x, y \leqq 1$ and let

$$\pi: 0 = x_0 < x_1 < \cdots < x_m = 1, \qquad \pi': 0 = y_0 < y_1 < \cdots < y_n = 1,$$

define a rectangular partition of this square, let $S(x, y)$ denote the bicubic spline function (8.1) interpolating $f(x, y)$ on this partition.

Since, for $i = 0, \cdots, m$, $S(x_i, y)$ is the cubic spline that interpolates $f(x_i, y)$ on π', one has from §4 that

$$\left| \frac{\partial^r}{\partial y^r} (S(x_i, y) - f(x_i, y)) \right| = O(|\pi'|^{4-r}), \qquad i = 0, \cdots, m. \tag{9.1}$$

Similarly, since, for $i = 0, m$, $S_x(x_i, y)$ is the cubic spline that interpolates $f_x(x_i, y)$ on π', one has

$$\left| \frac{\partial^{r+1}}{\partial y^r\, \partial x} (S(x_i, y) - f(x_i, y)) \right| = O(|\pi'|^{4-r}). \tag{9.2}$$

Now let $y \in [0, 1]$, and let $t(x)$ be the spline function that interpolates $f(x, y)$ on π. Then

$$\left| \frac{\partial^r}{\partial x^r} f(x, y) - t^{(r)}(x) \right| = O(|\pi|^{4-r}). \tag{9.3}$$

The difference between $t(x)$ and $S(x, y)$ is a spline function, hence can be written as $t(x) - S(s, y) = \sum_{i=1}^{n-1} a_i C_i(x) + p(x)$, where the $C_i(x)$ are the cardinal functions of §4, and $p(x)$ is a cubic polynomial. By (9.1) and (9.2),

$$|p(x)| = O(|\pi'|^4),$$

hence, by (9.1),

$$\begin{aligned} a_i = t(x_i) - S(x_i, y) - p(x_i) &= (f(x_i, y) - S(x_i, y)) - p(x_i) \\ &= O(|\pi'|^4). \end{aligned}$$

Since the $C_i(x)$ are uniformly bounded if M_π, $M_{\pi'}$ are suitably bounded (cf. §4), it follows that

$$|f(x, y) - S(x, y)| = O(|\pi|^4 + |\pi'|^4). \tag{9.4}$$

10. **General spline interpolation.** The preceding schemes have natural generalizations in two directions: to splines of higher degree and to functions of n variables.

To interpolate given values $f(x_i)$, $i = 0, \cdots, n$ by a spline function of odd degree $2m - 1$, $m > 2$, one proceeds much as in §3. One takes the quantities $f^{(j)}(x_i)$, $j = 1, \cdots, m - 1$, as unknowns to be determined from the conditions that $f^{(j)}$ be continuous across the joints for $j = m, \cdots, 2m - 2$. This results in a system of equations whose matrix is block-tridiagonal, each block being an $(m - 1) \times (m - 1)$ matrix. Given $f^{(j)}(x_0), f^{(j)}(x_n)$, $j = 1, \cdots, m - 1$, at the two endpoints, this system has a unique solution. With the $f^{(j)}(x_i)$ determined, $j = 0, \cdots, m - 1$, the $2m$ coefficients of the $(2m - 1)$st degree polynomial in each interval are then quickly computed.

An alternative approach makes use of the existence of a basis $M_i(x)$, $i = -(m - 1), \cdots, n + (m - 1)$, for the set of $(2m - 1)$st degree spline functions with joints at x_j, $j = 1, \cdots, n$, such that $M_i(x) \equiv 0$ for $x \notin [x_{i-m}, x_{i+m}]$ (this condition being properly modified for $i < m$ and $i > n - m$). The linear system which results is smaller than the system discussed earlier; its matrix is $(2m - 1)$-diagonal. This approach is equally well-suited to interpolation by spline functions of even degree. In this case, one follows Schoenberg [13] and puts the joints of the interpolating spline *between* the given data points. In the case of parabolic splines, the choice of the midpoint between given data points leads to a linear system with diagonally dominant tridiagonal matrix.

The generalization to functions of n variables follows the pattern of bicubic spline interpolation outlined earlier. The interpolating function becomes a tensor product of n one-dimensional interpolating functions, the interpolation conditions become correspondingly tensor products of n one-dimensional interpolation conditions.

One can also use, in any "hyper-rectangle", tensor products of spline functions of different degrees in different coordinate directions.

D. Approximation by Spline Functions

11. **Interpolation and approximation.** In applications, interpolation is commonly only a means to the end of obtaining good *approximations* to given functions or discrete data—and the latter themselves are often only approximate. Hence one naturally asks: are piecewise polynomial functions more suitable than (say) polynomial or rational functions, when it comes to approximation? As a background for this question, we recall some well-known facts.

First, as recalled already in §1, polynomials can be found which give arbitrarily good uniform approximations to any smooth function; but the degree may be extremely large. For fixed degree n, and functions of one variable, the Chebyshev Equioscillation Theorem [**4**, p. 149] shows that the best uniform approximation can also be obtained by Lagrange *interpolation* on a suitable mesh. Moreover one can today compute the points of this mesh systematically by the Remez algorithm [**8**, pp. 217–232]. The paper by Prof. Stiefel at this Symposium (p. 68) says more about their location.

From this standpoint, the most conspicuous advantage of spline interpolation over Lagrange interpolation, as a means of approximation, is that the error is less sensitive to mesh changes (see §4). A more subtle advantage is the smaller tendency to give *ripples*, whose occurrence with best uniform (Chebyshev) approximation seems indicated by the same Equioscillation Theorem. To avoid these, it seems likely that other criteria (such as best uniform approximation of derivatives) are needed, which would lead to additional computational problems.

However, it must be admitted that the preceding statements are conjectural, and not based on careful study. In practice, we have simply avoided polynomial approximation.

12. **Best approximation properties.** Actually, spline interpolation already has various "best approximation" properties, associated with its variational property of minimizing the positive quadratic functional $\int [y^{(h)}(x)]^2\,dx$. These have been derived by Walsh, Ahlberg, and Nilson [**17**], by one of us [**6**], and by Schoenberg ([**14**] and Indag. Math. **26** (1964), 155–163). For example let $s(x)$ be the spline function of odd degree $2k - 1$ which interpolates (is "cotabular" with) a given function $f(x) \in C^k[a, b]$ on the joints x_i of a given partition π of $[a, b]$. Then $s(x)$ is also the *best* approximation to $f(x)$ in the class of spline functions $S(a, b, \pi)$ with joints at the x_i, with respect to the pseudo-norm

$$\|e(x)\|^2 = \int_a^b [e^{(k)}(x)]^2\,dx. \tag{12.1}$$

It also leads to "best approximations" of linear functionals $L[f]$ such as $\int_0^1 f(x)\,dx$ or $f''(t)$ $(0 < t < 1)$ by linear combinations of the values of f at the joints x_i of a partition π: $0 \leqq x_1 < x_2 < \cdots < x_n \leqq 1$ (say). Following Sard [**12**], suppose that for fixed $m < n$, $L[p] = \sum_{i=1}^n a_i p(x_i)$ is exact for all polynomials $p(x)$ of degree m or less. Then, for $f(x) \in C^m[0, 1]$, defining $K(t)$ by

$$L[f] - \sum_{i=0}^n a_i f(x_i) = \int_0^1 K(t) f^{(m)}(t)\,dt, \tag{12.2}$$

the approximation $A[f] = \Sigma\, a_i f(x_i)$ to $L[f]$ is called "best" if it minimizes $\int_0^1 [K(t)]^2\,dt$. Schoenberg [**14**] has shown that if $S_m(x)$ denotes the interpolating (generalized) spline function of degree $2m - 1$, then the functional $A[f] = L[s_m[f]]$ is the "best" approximation (in this sense) to a wide class of linear functionals L.

As an application, consider the error of approximation to $\int_0^1 f(x)\,dx$ by

$\int_0^1 S(x)\,dx$, where $f(x) \in C^4$. This quadrature formula was first proposed by Holladay,[11] and later shown by Schoenberg [13] to be identical with Sard's best quadrature formula [12]. For simplicity, assume that π is uniform, $|\pi| = h$. Then it follows at once that

$$\int_0^1 f(x)\,dx = \int_0^1 S(x)\,dx + O(h^4), \tag{12.3}$$

the same order of accuracy as one gets by using the simplest Hermite-type quadrature formula in each of the n intervals of the partition. Note that in both cases (for uniform π), the value of the integral can be computed from $f(x_i)$, $i = 0, \cdots, n$, and $f'(0), f'(1)$.

Such results are valuable from a computational standpoint, since they permit the approximation of various functionals from the coefficients of interpolating splines computed by the schemes of §3.

But it should be pointed out that they are not restricted to spline functions, but apply to orthogonal projections generally.[12] Thus, for $m \geqq 1$, let $H^{(m)}$ be the Hilbert space defined on the functions $f \in C^{(m)}[0, 1]$ with absolutely continuous $(m-1)$st and square integrable mth derivative by the inner product

$$(f, g) = \int_0^1 f^{(m)}(x) g^{(m)}(x)\,dx + \sum_1^n L_i(f) L_i(g), \tag{12.4}$$

where the L_i are any n linearly independent linear functionals, linearly independent over the set of $(m-1)$st degree polynomials; let $\phi_j \in H^{(m)}$ be such that $L_j[f] = (f, \phi_j)$ for all $f \in H^{(m)}$, $j = 1, \cdots, n$. Let Φ be the subspace of $H^{(m)}$ spanned by the ϕ_i, and let $P = P_\phi$ be the orthogonal projection of $H^{(m)}$ onto Φ.

Then if $L[f] = (f, \psi)$ is any bounded linear functional on $H^{(m)}$, and $\bar{\psi}$ is the "best approximation" to ψ as an element of $H^{(m)}$, then the best approximation to $L[f]$ in the usual operator norm is $(f, \bar{\psi}) = \bar{L}[f]$. Moreover, for all $f \in H^{(m)}$, $\bar{L}[f] = (fP, \psi) = L[fP]$. Finally fP is the unique element of Φ which satisfies $L_i[fP] = L_i[f]$, $i = 1, \cdots, n$.

In the special case that the L_i are evaluations of $f(x)$ or some derivative $f^{(k)}(x)$ at points x_j, the space Φ consists of *piecewise polynomial* functions of degree $2m - 1$.

Finally, these facts remain true if D^m in (12.4) is replaced by other linear differential operators. Cf. also Golomb and Weinberger [8, pp. 117–190].

13. **Data smoothing.** The preceding results, however sharp, are far from answering all the basic questions which arise in fitting *approximate data* intended to represent (say) real car body surfaces or plane sections thereof. Such discrete data always involve random errors; hence they must be *smoothed* (or "graduated") without essential loss of accuracy.

[11] J. C. Holladay, Math. Tables Aids Comp. 11 (1957), 233–243. We have assumed here as given also $f'(0)$ and $f'(1)$.

[12] This observation is due to C. de Boor and R. E. Lynch [26].

This is a very deep problem. As is made clear in [19, §151], "the problem of graduation belongs essentially to the mathematical theory of probability". One must balance the *a priori* expectation of "smoothness" against the expectation that the error in the data (or their statistical deviation from the "true" mean) will be small.

In practice, this can often be achieved by considering an n-parameter family of smooth *approximating functions* $f(x, a_1, \cdots, a_n)$, where n is a small fraction of the number of data points $u(x_i) = u_i$, and minimizing the (suitably weighted) mean square deviation

$$\sum_{i=1}^{I} w_i |u_i - f(x_i, a_1, \cdots, a_n)|^2, \qquad w_i > 0. \tag{13.1}$$

In some cases, some form of Chebyshev approximation might be more accurate, but the computational simplicity and generality of *least squares* approximation for *linear* families of functions

$$f(x, a_1, \cdots, a_n) = \sum_{j=1}^{n} a_j \phi_j(x) \tag{13.2}$$

seems to make it the best bet for most applications.

For plane curves representing smooth sections of car bodies, we have found the use of formulas (13.1)–(13.2) with spline functions of degrees two, three, and four generally satisfactory. The problem of balancing smoothness against closeness of fit involves the problem of choosing the right number of joints. Since the closeness of fit depends on the location of the joints as well as on their number, one also has the *non-linear* problem of deciding on the *optimal location* of joints, for a given degree of spline (say).

We regard the determination of objective criteria for deciding this, and of mathematical methods for computing the location of optimal joints given such a criterion, as one of the most important problems concerning smoothing. We have used several criteria, including the following:

(i) For $w_i \equiv 1$, find the set of joints which minimizes (13.1),

(ii) For $w_i \equiv 1$, find the set of joints which minimizes

$$\min_{a_1, \cdots, a_n} \max_i |u_i - f(x_i, a_1, \cdots, a_n)|.$$

The error tends to oscillate in sign for both criteria, but we have not developed a rigorous theory for either of them.

A few remarks about the preceding problem may be of interest. To solve it objectively, one must clearly specify a measure N of error and a measure S of smoothness. For some "smoothing parameter" $\varepsilon > 0$, one can then try to minimize $\varepsilon N(u - f) + S(f)$ within the class of functions f, by proper choice of $a = (a_1, \cdots, a_n)$. One measure of smoothness for *discrete* data, proposed in [19], is provided by a sum of squares of the third divided differences; the continuous analog would consist in setting $S(f) = \int f'''^2\,dx$, as proposed by Schoenberg [13]. Other criteria have been proposed by Quade and Collatz, Lanczos, and Bizley.[13] For example (Lanczos), in trigonometric interpolation, one may know from differentiability considerations that the Fourier coefficients should die off like n^{-k}. One can then truncate Fourier series beginning at some points where this fails to hold.

Unfortunately, it seems impossible to combine the geometric desideratum of invariance of the approximating scheme under rotation (and translation) with the algebraic desideratum of linearity.

The problem of interpolating between smoothed "scans" along plane sections, so as to achieve a satisfactory smooth surface, is evidently related in nature but even more complicated. We shall not discuss it here.

E. Applications to Integral and Differential Equations

14. **Integral equations.**[14] Spline functions seem ideally suited to the approximate solution of Fredholm integral equations with *smooth kernels* $K(x, y)$, such as

$$f(x) = \phi(x) + \lambda \int_0^1 K(x, y) f(y)\, dy. \tag{14.1}$$

Relative to any partition π of the interval [0, 1] into n segments, one can construct by cubic spline interpolation and bicubic spline interpolation excellent approximations to $\phi(x)$ and $K(x, y)$, respectively—it being understood that π is applied to *both* independent variables ($I = J = n$).

When this is done, the right side of (14.1) is then replaced by an inhomogeneous linear operator $L[f]$ whose range is contained in the $(n + 3)$-dimensional subspace of (cubic) spline functions on [0, 1] with mesh π. Moreover, writing $L[f] = S[\phi] + \lambda K_s[f] = g$, the values of $g(x_i)$, $x_i \in \pi$, and of $g'(0), g'(1)$ are easily computed from those of $S[\phi]$ and $K_s[f]$, integrating in closed form polynomials (of degree 6 or less) in each segment.

This reduces (14.1) to an equation

$$f = S[\phi] + \lambda K_s[f] \tag{14.2}$$

which is equivalent to a system of linear *algebraic* equations in $n + 3$ variables, whose eigenvalues and solutions can be computed by standard methods. The results of §9 suggest that the error will be $O(n^{-3})$.

15. **Sturm-Liouville systems.** Another promising area of application for piecewise polynomial functions of degree k and class C^r is to the approximate numerical solution of boundary value problems. For functions of one variable, we will consider primarily the case $k = 3$, $r = 2$ of cubic *spline* functions, emphasizing applications of the concept of "best approximation" (§12) where possible.

The simpler case $k = 1$, $r = 0$ of "polygon" (piecewise linear) functions is classic.[15] Since polygon functions are defined by a *local* interpolation formula, their use is equivalent to *finite difference methods*, and is generally subsumed under the latter. Although there are many papers relevant to this case, they are not very relevant to the use of splines.

The use of spline functions for this purpose seems to have been first suggested in **[3]**. Here it was noted that one can compute approximate eigenfunctions and eigenvalues of Sturm-Liouville systems by applying the Rayleigh-Ritz-Galerkin method to the subspace $S = S(\pi)$ of (cubic) spline functions with a given mesh

$$\pi:\ 0 = x_0 < x_1 < \cdots < x_{N+1} = 1.$$

Example 1. Consider the Rayleigh quotient

$$R[f] = -J_1[f]/J_0[f], \tag{15.1}$$

where

$$J_0[f] = \int_0^1 [f(x)]^2\, dx, \qquad J_1[f] = \int_0^1 [f'(x)]^2\, dx = -\int_0^1 f(x)\, D^2[f(x)]\, dx. \tag{15.2}$$

The eigenfunctions of the Sturm-Liouville system $u'' + \lambda u = 0$, $u(0) = u(1) = 0$, are those functions satisfying the endpoint conditions, for which $\delta R = 0$.

Since the eigenfunctions $\sin m\pi x$ satisfy

$$|S_m(x) - \sin m\pi x| = O(h^4), \tag{15.3}$$

$$|S'_m(x) - m\pi \cos m\pi x| = O(h^3), \tag{15.3'}$$

the error in (15.3′) being $O(h^4)$ at mesh-points, it follows that the Rayleigh-Ritz-Galerkin method applied to the subspace $S(N)$ of (cubic) spline functions on a uniform mesh and satisfying $S(0) = S''(0) = S(1) = S''(1) = 0$ should give approximate eigenvalues μ_m (and eigenfunctions $w_m(x)$) in error by $O(h^3)$ or less, for any fixed m as $h \downarrow 0$.

However, we prefer to view the problem somewhat differently, so as to bring out the connection with the ideas of §12. For any given inner product $(f, g)_l$ defined on a function space which contains the subspace $S = S(N)$ just defined, let P_l be the *orthogonal projection* onto S. We observe (as did Galerkin) that for $f \in S$

$$R[f] = \int_0^1 f(x) P[D^2[f]](x)\, dx \Big/ \int_0^1 f^2(x)\, dx, \tag{15.4}$$

provided that P is symmetric with respect to the inner product $(f, g)_0 = \int f(x) g(x)\, dx$; hence the approximate eigenvalues and eigenfunctions are those of the linear operator PD^2, considered as an operator on S.

[13] See W. Quade and L. Collatz, S.-B. Preuss. Akad. Wiss. (Math-Phys. Kl.) **30** (1938), pp. 29–38; C. Lanczos, *Applied Analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1956, 321–344; M. T. L. Bizley, J. Inst. Actuaries Sept. (1958), 125–165.

[14] The considerations of §14 occurred independently to Prof. I. Schoenberg, with whom we have had several stimulating conversations.

[15] See for example R. Courant, Bull. Am. Math. Soc. **49** (1948), 1–23.

More generally, for any projection P onto S, we define the *relativization* of D^2 to S defined by P as the operator

$$E[f] = P[D^2[f]]. \tag{15.5}$$

We then ask: how well do the eigenfunctions and eigenvalues of E approximate those of D^2? We now answer this question exactly for three projections P_0, P_1, P_2. The first two are the *orthogonal* projections associated with the inner products J_0 and J_1 respectively:

$$(f, g)_0 = \int_0^1 f(x)g(x)\,dx \quad \text{and} \quad (f, g)_1 = \int_0^1 f'(x)g'(x)\,dx. \tag{15.6}$$

The projection P_2 is that associated with the inner product $\int_0^1 f''(x)g''(x)\,dx$ and defined by spline interpolation itself; cf. §12.

It is shown in Appendix B that, for these three projections, the orders of accuracy obtained in computations of the eigenvalues are $O(h^{6-2l})$ for $l = 0, 1, 2$. In all three cases, the approximate eigenfunctions are $P_2(\sin m\pi x)$, hence their order of accuracy is $O(h^4)$.

16. **Self-adjoint elliptic equations.** A fascinating field for future research concerns the usefulness of piecewise polynomial functions in describing approximate solutions of partial differential equations. We consider now the case of self-adjoint elliptic equations, with special reference to the Poisson equation

$$-\nabla^2 u = S(x, y). \tag{16.1}$$

Since solutions of such equations minimize suitable quadratic functionals, the Rayleigh-Ritz-Galerkin method can presumably be applied in much the same way as in §15. See example 2 below.

This idea has already been applied by Synge [**23**, pp. 168 ff.] to piecewise linear ("pyramidal") functions.[16] His results are analogous to those of Pólya and Szegö,[17] but the latter made much more extensive use of analytic functions and analytic methods (especially Steiner symmetrization).

In this connection, one should also mention an ingenious method of finding *analytic approximations* to solutions of the Dirichlet problem for $\nabla^2 u = 0$ proposed by S. Bergman,[18] and applied by him to many other boundary value problems of elliptic type. It is classic [**16**, pp. 36, 45] that any harmonic function can be approximated uniformly in any compact connected domain by harmonic polynomials. Bergman's method consists in computing which harmonic polynomial of given degree most closely fits the boundary conditions (i.e., best approximates them in some sense, mean square or Chebyshev).

[16] It would be interesting to compare results obtained using his "square pyramid F-vectors", with results using piecewise *bilinear* functions instead.

[17] G. Pólya and G. Szegö, *Isoperimetric inequalities in mathematical physics*, Annals of Math. Study No. 27, Princeton Univ. Press, Princeton, N.J., 1951.

[18] S. Bergman, Quar. Appl. Math. **5** (1947), 69–81; Proc. VI Symposium Applied Math., Amer. Math. Soc., Providence, R.I., 1956, pp. 11–29, and refs. given there.

However, this method seems less well adapted to computing machines than *difference methods*. These define what are sometimes called *discrete harmonic functions*, satisfying some (typically, 5-point) analog of $\nabla^2 u = 0$ or (16.1), and appropriate boundary conditions. There is a large literature on such functions[19], which we will not discuss here.

Instead, we will simply point out that the approximate solutions obtained in this way *cannot* be smoothly interpolated by harmonic polynomials, and so it seems reasonable to expect that interpolation can most effectively be made using piecewise polynomial functions, fitted by local bicubic or bicubic spline interpolation, for example.

It would be interesting to know whether any approximate solutions obtained from difference approximations in this way coincided with approximate solutions obtained from the same class of piecewise polynomial trial functions by the Rayleigh-Ritz-Galerkin method. For example, does the use of piecewise bilinear functions and a square or rectangular mesh lead to the standard[20] 5-point or 9-point formula for $\nabla^2 u = 0$?

To conclude this section, we shall give a discussion of the eigenvalues of the Helmholtz equation in a square.

EXAMPLE 2. Consider the Helmholtz equation $\nabla^2 u + \lambda u = 0$ in the unit square $0 \leq x, y \leq 1$, with boundary condition $u = 0$. We apply the Rayleigh-Ritz-Galerkin method to the subspace of bicubic spline functions having joints on a square grid (mesh-length $h = 1/N$ in both coordinates), and satisfying the "free edge" condition $\partial^2 u/\partial n^2 = 0$ (and hence $u_{xxyy} = 0$ at the four corners). The variables are "separable", and so the approximate eigenfunctions are $w_i(x)w_j(y)$, with approximate eigenvalues $\mu_i^l + \mu_j^l$, where μ_i^l is the corresponding eigenvalue of $w_i(x)$ under P_l (cf. Example 1 in §15).

17. **Cauchy problems.** We now consider possible applications of the preceding ideas to initial value problems for partial differential equations or *Cauchy problems*. In this, we follow the general approach of [**25**], restricting attention for simplicity to Cauchy problems whose exact solutions define C_0-semigroups G of bounded linear transformations $T_t = \exp(tL)$ of some Hilbert space $\mathfrak{H}$, with infinitesimal generator L.

For any finite-dimensional or other closed subspace S of $\mathfrak{H}$ in the domain of L, define the *orthogonal projection* G_S of G onto S as follows. Let P be the orthogonal projection of $\mathfrak{H}$ onto S, with null-space $S^{\perp}$; then $P[L[u]]$ is a bounded linear transformation L_S of S. We let G_S be the C_0-semigroup acting on S generated by PL. Equivalently, we can define the projection of T_t onto S as $\lim_{n\to\infty} (PT_{t/n})^n$.

For example, let G be the C_0-semigroup on the Hilbert space $L_2(0, 1)$ associated

[19] [**22**] and refs. given there; Duffin, etc.

[20] [**24**, Chapter 6]; R. Esch, Annals of the Harvard Computation Laboratory, Vol. 31, pp. 84–102, Harvard Univ. Press, Cambridge, Mass., 1962.

with the heat equation $u_t = u_{xx}$ and the boundary conditions $u(0) = u(1) = 0$. Let $S = S_N$ be the set of functions expressible as *truncated sine series* of the form

$$u(x) = a_1 \sin x + a_2 \sin 2x + \cdots + a_N \sin Nx. \tag{17.1}$$

In this example, since the subspace S_N is invariant under L, the orbits of G_S also define exact solutions of the given (mixed) boundary value problem.

The preceding property holds in many other examples. Thus, let L be any linear differential operator with constant coefficients on any periodic or infinite domain. Then the subspaces defined by truncated Fourier series, and by truncated Fourier integrals $\int_{-\pi N}^{\pi N} e^{iq\cdot x}\phi(q)\,dq$, are invariant under L. On the other hand, these subspaces are also the ranges of the *projection operators* defined by the schemes of trigonometric interpolation **[7]** and cardinal interpolation **[18]**, already described in §1. As is explained in **[25]**, each interpolation scheme J defines a projection operator on continuous functions $f(x)$, which maps any f into $J[\tau[f]]$, where τ is the *tabulation* operator. It follows that, knowing the initial error $\|u_0 - J[\tau[u_0]]\|$ (which expresses the loss of information caused by identifying "cotabular" functions), the error $\|u(t) - u_J(t)\|$ at any later time t is at most $\|T_t\| \cdot \|u_0 - J[\tau[u_0]]\|$.

The statements of the preceding paragraph apply specifically to trigonometric and cardinal interpolation, for which it should be remembered that the projection $P[u] = J[\tau[u]]$ is not orthogonal in the usual L_2-norm. We now make a preliminary study of the analogous situation as regards cubic spline interpolation.

Let S be the N-dimensional subspace of cubic splines defined in Example 1 (§15), and let $\mathfrak{H}$ be defined by one of the inner products $(f, g)_l$, $l = 0, 1$, of (15.6). Then, for the heat equation $u_t = u_{xx}$ and the general initial condition $u(x, 0) = \Sigma\, a_i w_i(x)$, where the $w_i(x)$ are the approximate eigenfunctions of Example 1, we have $u(x, t) = \Sigma\, a_i e^{-\mu_i^l t} w_i(x)$, where μ_i^l is the eigenvalue of $w_i(x)$. Hence G_S is a semigroup of diagonal matrices $\bar{E} = \delta_{ij} e^{-\mu_i^l t}$ relative to this basis. Numerical values of μ_i^l can be computed from the formulas of Appendix B.

Likewise, let T be the N^2-dimensional subspace of bicubic spline functions in the square, defined in Example 2 (§16), and consider the analogous inner products

$$(f, g)_0 = \int_0^1\int_0^1 f(x, y) g(x, y)\,dx\,dy \tag{17.2}$$

and

$$(f, g)_1 = \int_0^1\int_0^1 [f_x g_x + f_y g_y]\,dx\,dy. \tag{17.3}$$

Then, relative to the eigenfunctions $w_i(x)w_j(y)$ of this example, the semigroup defined by the heat equation is defined by the "tensor product" formula

$$u(x, y, t) = \sum a_{ij} e^{-(\mu_i^l + \mu_j^l)t} w_i(x) w_j(y). \tag{17.4}$$

Appendix A. Piecewise Bicubic Interpolation in Rectangular Polygons

1. **Interpolation of class C^1.** Let R be any connected rectangular polygon, whose sides all lie on mesh-lines $x = x_i$, $y = y_j$, of a fixed rectangular mesh. As in §3 and **[3**, §§1–2**]**, we have the following result.

Lemma 1. *Let $u_{ij} = u(x_i, y_j)$ be given at all (interior and boundary) mesh-points of R, let $p_{ij} = u_x(x_i, y_j)$ be given on the sides $x = x_i$ of R which are parallel to the y-axis (except at reentrant corners), and let $q_{ij} = u_y(x_i, y_j)$ be given on the perpendicular sides $y = y_j$. These data are compatible with one and only one network of cubic spline functions $u(x, y_j) = f_j(x)$ and $u(x_i, y) = g_i(y)$ on the mesh-lines of R.*

We next recall **[5**, Theorem 2**]**, which states

Lemma 2. *There exists one and only one bicubic polynomial*

$$u(x, y) = \sum_{m=0}^{3}\sum_{n=0}^{3} \alpha_{mn} x^m y^n \tag{A.1}$$

which assumes given values of u, u_x, u_y, and u_{xy} at the four corners of a given rectangle.

Lemmas 1 and 2 have the following consequence.

Lemma 3. *Let u, u_x, u_y, and u_{xy} be given at the vertices of two adjacent rectangles R_1, R_2 having a common side $y = y_j$. Then there is one and only one function $u(x, y)$ of class C^1 on $R = R_1 \cup R_2$, which assumes the given values and is bicubic in each R_j, $j = 1, 2$. For this function, u_{xy} and u_{xxy} are also continuous.*

Proof. By Lemma 2, there is only one such function. But for this function, $u(x, y_j)$ and $u_y(x, y_j)$ have the same values in R_1 as in R_2 at the endpoints of $y = y_j$; from these their other values are obtained by spline interpolation; hence they are the same identically. The continuity of u_x, u_{xx}, $\cdots$ and u_{xy}, u_{xxy}, $\cdots$ across the common edge $y = y_j$ follows.

Theorem 1. *In Lemma 1, given $s_{ij} = u_{xy}(x_i, y_j)$ at all mesh-points, there exists a unique piecewise bicubic function $u(x, y)$ of class C^1 in R, for which the mesh-lines are spline curves, and which satisfies the data of Lemma 1. For this $u \in C^1(R)$, $s(x, y) = u_{xy}$ is also continuous.*

To obtain a reasonably accurate and very simple scheme of piecewise bicubic interpolation of class C^1, it suffices therefore to use some fairly good local approximation to u_{xy} at mesh-points. For example, one might use

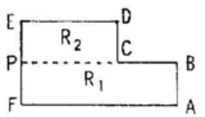
$$\sigma_{ij} = w\left[\frac{p_{i,j+1} - p_{i,j-1}}{y_{j+1} - y_{j-1}}\right] + (1 - w)\left[\frac{q_{i+1,j} - q_{i-1,j}}{x_{i+1} - x_{j-1}}\right], \tag{A.2}$$

where $p_{ij} = u_x(x_i, y_j)$, $q_{ij} = u_y(x_i, y_j)$, and $0 \leqq w \leqq 1$ (or, $0 \leqq w_{ij} \leqq 1$). Or, one might use at interior mesh-points

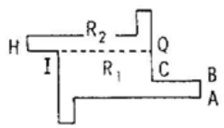
$$\text{(A.2')} \qquad \tau_{ij} = \frac{u_{i+1,j+1} - u_{i+1,j-1} + u_{i-1,j-1} - u_{i-1,j+1}}{(x_{i+1} - x_{i-1})(y_{j+1} - y_{j-1})},$$

as approximations to the "best" $s_{ij} = u_{xy}(x_i, y_j)$.

2. **Interpolation of class C^2.** The problem of devising a "well-set" piecewise bicubic interpolation scheme of class C^2 in a general (connected) rectangular polygon R is much more difficult. We will suppose below that u and its normal derivative are given and differentiable along the edges of R, and that the given values of the normal derivative are compatible in the sense that $(u_x)_y = (u_y)_x$ at external (i.e., non-reentrant) corners. Figures 1 and 2 illustrate two instructive cases.



Rectangular hexagon
FIGURE 1



Rectangular dodecagon
FIGURE 2

Let R be subdivided into two subregions R_1 and R_2 by a mesh-line $y = y_J$, $x_0 \leqq x \leqq x_M$, and suppose that common values of u and u_y are assumed along $y = y_J$ by functions $u_k(x, y)$ defined and of class C^1 in each R_k $[k = 1, 2]$ separately. Suppose also that $u_1(x, y)$ and $u_2(x, y)$ are both *piecewise bicubic*, i.e., satisfy (A.1) in each rectangle R_{ij} bounded by pairs of adjacent mesh-lines. Finally, suppose that the values of the $u_k(x, y)$ are defined by cubic spline functions along each mesh-line, so that u_{xx} is continuous on the mesh-lines $x = x_i$ and u_{yy} on the mesh-lines $y = y_j$.

Then, by Lemma 3, u_{xy} will be continuous across $y = y_J$, together with $u, u_x, u_{xx}, \cdots$ and $u_y, u_{xxy}, \cdots$. But u_{yy}, the second normal derivative, need not be continuous. Though we know of no method for making u_{yy} continuous, the jump in u_{yy} can usually be made negligible in practice by proceeding as follows.

Where possible, use *spline interpolation* to determine u_{xy} at reentrant corners. For example, in Figure 1, use spline interpolation in u_x on $\overline{EPF}$ to compute u_{xy} at P, then use spline interpolation in u_y on $\overline{PCB}$ to compute u_{xy} at C. One can then use *bicubic* spline interpolation in R_1 and R_2, to determine functions $u_k(x, y)$ of class C^2 in R_k $[k = 1, 2]$. By splicing these functions together, one obtains a piecewise bicubic spline function of class C^1 in $R_1 \cup R_2 = R$. We now examine more closely the behavior of the higher derivatives of $u(x, y)$ along the "seam" $\overline{PC}$. By construction (cf. §5), u_{yy} is continuous across $\overline{PC}$ (i.e., it assumes the same value on both sides) at *all mesh-points*, including P and C; moreover u_{xy} and u_{xx} assume the

same values at *all points*. Furthermore (§6), u_{yy} is obtained on *both* sides of $\overline{PC}$ by *spline interpolation* from the same u_{yy}, though generally different "endslopes" u_{yyx} at P and C. Due to the *stability* (§8) of spline interpolation, the difference between the values of u_{yy} along $\overline{PC}$ on the two sides will therefore be an *oscillating, exponentially damped* function as one goes away from the endpoints (for uniformly bounded mesh-length ratios $|\Delta x_i|/|\Delta x_{i'}| \leqq M_\pi$). Hence, if M_π is held fixed while the maximum mesh-length tends to zero, the values of u_{yy} will approach the same limit at all points other than reentrant corners.

In the rectangular polygon of Figure 2, no such simple procedure is possible. For expeditious computation, it seems best to assign (by local extrapolation) an approximate value to u_{xy} at C, and then to decompose R into rectangles by induction, using HIQ as one boundary; at worst, one has more than the minimum number of "seams" to consider.

Though we doubt the existence of a "stable" (i.e., uniformly bounded) procedure for finding a piecewise cubic function of class C^2 in a general rectangular polygon (e.g., in the L-shaped region of Figure 1), the following result may be of interest.

THEOREM 2. *Let the mesh-line $y = y_j$, $x_0 \leqq x \leqq x_M$ divide the rectangular polygon R into subregions R_1 and R_2. Let $u(x, y)$ be a piecewise bicubic function which is of class C^2 in R_1 and R_2 separately and on mesh-lines, and of class C^1 in R. Then $u(x, y)$ is of class C^2 in R if and only if $s_{ij} = u_{xy}(x_i, y_j)$ satisfies*

$$\text{(A.4)} \quad bs_{i,j+1} + 2(b + c)s_{i,j} + cs_{i,j-1} = 3\left[\frac{b}{c}(p_{i,j+1} - p_{i,j}) + \frac{c}{b}(p_{i,j} - p_{i,j-1})\right], \qquad i = 0, M,$$

where $b = y_j - y_{j-1}$ and $c = y_{j+1} - y_j$.

We omit the proof.

APPENDIX B. STURM-LIOUVILLE COMPUTATIONS

In order to compute the eigenfunctions and eigenvalues of $E_l = P_l D^2$, it is convenient to pick a basis $\{t_i\}_1^N$ of $S = S(N)$ and to consider the matrix representation $\bar{E}_l$ of E_l with respect to that basis. Its eigenvectors are the coefficients of the eigenfunctions of E_l with respect to $\{t_i\}_1^N$, and its eigenvalues μ_1^l are those of E_l.

For $l = 0, 1$, $P_l f$ satisfies

$$\text{(B.1)} \qquad (f, t_i)_l = (Pf, t_i)_l, \qquad i = 1, \cdots, N.$$

Hence, for $s = \sum_{i=1}^N \alpha_i t_i \in S$, with $E_l s = \sum_{i=1}^N \beta_i t_i$, $t_i'' = (D^2 t)_i$ satisfies

$$\sum_{i=1}^N \alpha_i (t_i'', t_j)_l = \sum_{i=1}^N \beta_i (t_i, t_j)_l;$$

therefore

(B.2) $\bar{E}_l = A_l^{-1}B_l, \qquad A_l = \{(t_i, t_j)_l\}, \qquad B_l = \{(t_i'', t_j)_l\}, \qquad l = 0, 1.$

We now choose a basis $\{t_i\}_1^N$ of S convenient for the computation of $\bar{E}_k$. Define

(B.3)
$$t_i(x) = (x - x_{i-1})_+^3 - 2(x - x_i)_+^3 + (x - x_{i+1})_+^3 + a_i x$$
$$= \begin{cases} a_i x, & x \leqq x_{i-1}, \\ a_i x + (x - x_{i-1})^3, & x_{i-1} \leqq x \leqq x_i, \\ b_i(x - 1) - (x - x_{i+1})^3, & x_i \leqq x \leqq x_{i+1}, \\ b_i(x - 1), & x_{i+1} \leqq x, \end{cases}$$

where $a_i = 6h^2(x_i - 1)$, $b_i = 6h^2 x_i$, $i = 1, \cdots, N$. This choice makes B_1 tri-diagonal, since $t_i''(x) \equiv 0$ for $x \notin (x_{i-1}, x_{i+1})$.

CASE $l = 1$. In fact, one computes that

$$B_1 = -6h^3(C + 6),$$

where C is the $N \times N$ tridiagonal matrix with general row $\{1, -2, 1\}$. With somewhat more effort, one computes that

$$A_1 = -\tfrac{3}{10}h^5(C + 30 + 120C^{-1}).$$

Consequently

(B.4)
$$\bar{E}_1 = \frac{20}{h^2}\frac{C(C + 6)}{C^2 + 30C + 120}.$$

But this implies that the eigenvectors of $\bar{E}_1$ are those of C, viz., $\{\sin ijh\pi\}_{j=1}^N$. Hence if $\{\lambda_i\}_1^N$ denotes the corresponding set of eigenvalues of C ($\lambda_i) = 2(\cos ih\pi - 1)$, then the eigenvalues μ_i^1 of $\bar{E}_1$ and E_1 are

(B.5)
$$\mu_i^1 = \frac{20}{h^2}\frac{\lambda_i(\lambda_i + 6)}{\lambda_i^2 + 30\lambda_i + 120},$$

while the eigenfunctions w_i^1 of E_1 are

(B.6)
$$w_i^1(x) = \sum_{j=1}^{N} t_j(x) \sin ijh\pi.$$

CASE $l = 0$. One proceeds just as in the case $l = 1$. The matrix B_0 is $-A_1$. The matrix A_0 turns out to be

$$A_0 = \frac{36}{7!}h^7(C + 126) + 12h^7C^{-1} + 36h^7C^{-2}.$$

Hence the eigenvalues μ_i^0 of $\bar{E}_0$ and E_0 are

(B.7)
$$\mu_i^0 = \frac{42}{h^2}\frac{\lambda_i(\lambda_i^2 + 30\lambda_i + 120)}{\lambda_i^3 + 126\lambda_i^2 + 1680\lambda_i + 5040},$$

while the eigenfunctions w_i^0 of E_0 are again those of E_1.

In fact, one verifies that

(B.8) $w_i^0 = w_i^1 = h^3(1 + 6\lambda_i^{-1})P_2(\sin i\pi x), \qquad i = 1, \cdots, N,$

where P_2 is the projection associated with spline interpolation, as defined in §15.

CASE $l = 2$. In this case, one makes use of the identity valid for $f \in S$:

$$\frac{h^2}{6}(f''(x_{i-1}) + 4f''(x_i) + f''(x_{i+1})) = f(x_{i-1}) - 2f(x_i) + f(x_{i+1}).$$

This gives

$$E_2 = \frac{6}{h^2}\frac{C}{C + 6},$$

so

(B.9)
$$\mu_i^2 = \frac{6}{h^2}\frac{\lambda_i}{\lambda_i + 6}, \qquad i = 1, \cdots, N,$$

while the eigenfunctions of E_2 are again those of E_1 (and E_0).

In all three cases, the approximation to the eigenfunctions is, therefore, $O(h^4)$. The approximations to the eigenvalues, on the other hand, vary in order of accuracy. Let ν_i denote the eigenvalues of D^2, i.e.,

$$\nu_i = -(i\pi)^2, \qquad i = 1, 2, \cdots;$$

then one computes

$$\mu_i^2 = \nu_i\left(1 - \frac{2}{4!}h^2\nu_i + \frac{2}{6!}h^4\nu_i^2 - \cdots\right),$$

$$\mu_i^1 = \nu_i\left(1 + \frac{1}{6!}h^4\nu_i^2 - \frac{12}{8!}h^6\nu_i^3 + \cdots\right),$$

$$\mu_i^0 = \nu_i\left(1 - \frac{4}{3}\frac{1}{8!}h^6\nu_i^3 + \cdots\right).$$

Hence the application of the Rayleigh-Ritz-Galerkin method to the space of cubic splines with uniformly spaced joints gives an approximation to the eigenvalues of D^2 of order $O(h^6)$. The following table gives $-\mu_i^l$ for $l = 0, 1, 2$, $N = 1, 2, 3$, and $i = 1, 2$.

	$l = 2$	$l = 1$	$l = 0$	Exact	
$N = 1$	12.0	10.0	9.8824	9.86960438	$i = 1$
$N = 2$	10.8	9.8901	9.870300		
$N = 3$	10.39	9.8755	9.869706		
$N = 2$	54.0	41.54	39.9512	39.478416	$i = 2$
$N = 3$	48.0	40.00	39.5304		

References

1. J. H. Ahlberg and E. N. Nilson, *Convergence properties of the spline fit*, J. Soc. Indust. Appl. Math. **11** (1963), 95–104.
2. G. Birkhoff and C. de Boor, *Error bounds for cubic spline interpolation*, J. Math. Mech. **13** (1964), 827–835.
3. G. Birkhoff and H. L. Garabedian, *Smooth surface interpolation*, J. Math. and Phys. **39** (1960), 258–268.
4. P. J. Davis, *Interpolation and approximation*, Blaisdell, New York, 1963.
5. C. de Boor, *Bicubic spline interpolation*, J. Math. and Phys. **41** (1962), 212–218.
6. C. de Boor, *Best approximation properties of spline functions of odd degree*, J. Math. Mech. **12** (1963), 747–750.
7. D. Jackson, *The theory of approximation*, Amer. Math. Soc., Providence, R.I., 1930.
8. R. Langer, editor, *On numerical approximation*, Univ. of Wisconsin Press, Madison, Wisc., 1959.
9. A. E. H. Love, *The mathematical theory of elasticity*, 4th ed., Cambridge Univ. Press, London, 1934.
10. C. Runge, *Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten*, Zeits. Math. Phys. **46** (1901), 224–243.
11. H. Rutishauser, *Bemerkungen zur glatten Interpolation*, ZAMP **11** (1960), 508–513.
12. A. Sard, *Best approximate integration formulas; best approximation formulas*, Amer. J. Math. **71** (1949), 352–357; see also, *Linear approximation*, Math. Surveys No. 9, Amer. Math. Soc., Providence, R.I., 1963.
13. I. J. Schoenberg, *Contribution to the problem of approximation of equidistant data by analytic functions*. Parts A and B, Quar. Appl. Math. **4** (1946), 45–99, 112–141.
14. I. J. Schoenberg, *Spline interpolation and best quadrature formulae*, Bull. Amer. Math. Soc. **70** (1964), 143–148.
15. H. Thacher, editor, *Numerical properties of functions of more than one independent variable*, Annals New York Acad. Sci. **86** (1960), 677–874.
16. J. L. Walsh, *Interpolation and approximation by rational functions in the complex domain*, Amer. Math. Soc., Providence, R.I., 1956.
17. J. L. Walsh, J. H. Ahlberg, and E. N. Nilson, *Best approximation properties of the spline fit*, J. Math. Mech. **11** (1962), 225–234.
18. E. T. Whittaker, *On the functions which are represented by the expansions of the interpolation-theory*, Proc. Roy. Soc. Edinburgh (1915), 181–194.
19. E. T. Whittaker and G. Robinson, *The calculus of observations*, 4th ed., Blackie, London, 1944.
20. I. J. Schoenberg, *On trigonometric spline interpolation*, J. Math. Mech. **13** (1964), 795–826.
21. G. Birkhoff, H. Burchard, and D. Thomas, *Nonlinear interpolation by splines, pseudosplines, and elastica*, Res. Publ. General Motors Research—468 (1965).
22. B. van der Pol, Appendix IV of M. Kac, *Probability and related topics in the physical sciences*, Interscience Press, New York, 1959.
23. J. L. Synge, *The hypercircle in mathematical physics*, Cambridge Univ. Press, London, 1957.
24. R. S. Varga, *Matrix iterative analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1962.
25. G. Birkhoff, *Boundary value problems, function spaces, and computing*, Trans. 10th Conference of Army Mathematicians, U.S.A.
26. C. de Boor and R. E. Lynch, *Generalized splines and their minimum properties*, unpublished MS.
27. A. H. Fowler and C. W. Wilson, *Cubic spline, a curve fitting routine*, Report Y-1400, Oak Ridge, 1963.
28. D. H. MacLaren, *Formulas for fitting a spline curve through a set of points*, Boeing Appl. Math. Report 2, 1958.