JOURNAL OF APPROXIMATION THEORY 2, 85–96 (1969)

Computational Methods for Best Spline Function Approximation

R. E. ESCH AND W. L. EASTMAN

Boston University, Boston, Massachusetts, 02215
and
Sperry Rand Research Center, Sudbury, Massachusetts, 01776

1. INTRODUCTION

Suppose it is desired to approximate in the Chebyshev or L_∞ norm an arbitrary continuous function $f(x)$ over $\alpha \leqslant x \leqslant \beta$ by a spline function $A(x)$ of degree n with $m-1$ interior joints or knots $u_1, \ldots, u_{m-1}$, where

$$\alpha = u_0 < u_1 \ldots < u_{m-1} < u_m = \beta.$$

The approximating function $A(x)$, then, where

$$A(x) \in C^{n-1} \tag{1}$$

is required to be a polynomial of degree not exceeding n within any interval $u_{k-1} \leqslant x \leqslant u_k$, $k = 1, 2, \ldots, m$. One representation for $A(x)$, displaying its linear dependence on $n+m$ parameters d_i and c_k, is

$$A(x) = \sum_{i=0}^{n} d_i x^i + \sum_{k=1}^{m-1} \frac{c_k}{n!}(x-u_k)_+^n, \tag{2}$$

where

$$(x)_+^p = \begin{cases} 0 & x < 0 \\ x^p & x \geqslant 0. \end{cases} \tag{3}$$

The quantity c_k gives the discontinuity in the nth derivative of $A(x)$ at the kth joint u_k.

Various generalizations are of interest. If two joints of $A(x)$ are allowed to coalesce, then a point at which

$$\frac{d^{n-2}A}{dx^{n-2}} \equiv A^{(n-2)}(x)$$

is discontinuous can arise. If the joints are to be regarded as free parameters, then it is necessary to close the set of admissible approximations $A(x)$ by adding such limits in order to guarantee the existence of best approximations.

On the other hand, one may be interested in a lower degree of continuity from the start. If (1) is replaced by $A(x) \in C^r$, then Eq. (2) is replaced by

$$A(x) = \sum_{i=0}^{n} d_i x^i + \sum_{s=0}^{n-r-1} \sum_{k=1}^{m-1} \frac{c_{s,k}}{(n-s)!}(x-u_k)_+^{n-s}. \tag{4}$$

Another type of generalization is made by Schumaker (1967, 1968), who replaces polynomials by extended complete Chebyshev systems.

In any of these cases $A(x)$ has a representation of the form

$$A(x; \mathbf{a}, \mathbf{u}) = \sum_{j=1}^{\nu} a_j \Phi_j(x; \mathbf{u}) \tag{5}$$

where, for example, for Eq. (2), $\nu = n + m$. If the joints $\mathbf{u}$ are regarded as given, we have a linear approximation problem where, however, the functions Φ_i do not form a Chebyshev system. If the joints are included among the parameters to be optimized, the approximation problem is nonlinear.

Schumaker (1968) has given a computational method for the fixed-joint problem based on an exchange process of the Remez variety. Such an approach, which assumes an error curve of "normal" form with $\nu + 1$ alternating extrema, suffers a disadvantage because of the non-Chebyshev nature of the approximating function; for some $f(x)$ the best approximations have fewer than $\nu + 1$ alternating extrema. Perhaps a greater nuisance in practice is that, when the joints are near their optimal locations, additional error oscillations appear.

Barrodale and Young (1967) and the authors (1967) have used a linear programming approach which, though closely related to exchange processes of the Remez variety, is not based explicitly on any characterization process, and thus avoids the above difficulties. The linear programming formulation is described in Section 2; it furnishes a basic subroutine used in computational attacks on the variable-joint problem discussed in Section 3.

Rice (1967) has given a full characterization theory for best spline approximation in the fixed-joint case. Schumaker (1967a, b) has independently treated this problem and also the free-joint problem where the joints are parameters to be optimized. Schumaker's characterization results for the free-joint case furnish a theoretical basis for the computational algorithms discussed in Section 3.

Powell (1966) has discussed the corresponding L_2 approximation problem. Also noteworthy is Lawson's treatment of piecewise approximation without continuity constraints at the joints [(1963), (1964)].

2. A Computational Method for the Fixed-Joint Case

Given the joints $\mathbf{u} = (u_1, \ldots, u_{m-1})$, an arbitrary continuous function $f(x)$, and the norm

$$\|\Phi(x)\| = \max_{\alpha \leqslant x \leqslant \beta} |\Phi(x)|, \tag{6}$$

we want to find that member of the admissible class of functions (5) which minimizes the error norm $\|f-A\|$; that is, we want a method to compute $\mathbf{a}^*$, where for any $\mathbf{a}$

$$\|f(x)-A(x;\mathbf{a}^*,\mathbf{u})\| \leqslant \|f(x)-A(x;\mathbf{a},\mathbf{u})\|.$$

As a first step, we agree to look only at a finite set of points $Y=(\xi_1,\xi_2,\ldots,\xi_N)$, where N might be, for example, 1000; that is, we replace the norm (6) by the finite point set norm

$$\|\Phi(x)\|_Y \equiv \max_{x\in Y} |\Phi(x)|.$$

The "discretization error" thereby introduced can be studied by the approaches of Rivlin and Cheney (1966), Rice (1964), or Shisha (1966); it is easy thereby to obtain *a posteriori* estimates of this error in terms of moduli of continuity of $f(x)$ and the computed $A(x)$. Alternatively (and always to be recommended in careful numerical work), an effective computational procedure is to solve one or more test problems on successively finer meshes and observe the convergence of the results. As a matter of computational experience, when the error curve is smooth, the discretization error will ordinarily be unimportant if there are at least half a dozen points in each loop of the error curve. If a variable mesh is used it should of course be finer in regions where the error curve is changing more rapidly.

The discretized problem can now be formulated as a linear programming problem:

$$\left.\begin{array}{l} \text{Minimize } \lambda \\ \text{subject to the } 2N \text{ inequalities} \\ \left.\begin{array}{l} \lambda-\left[f(\xi_i)-\sum_{j=1}^{\nu} a_j\Phi_j(\xi_i)\right]\geqslant 0 \\ \lambda+\left[f(\xi_i)-\sum_{j=1}^{\nu} a_j\Phi_j(\xi_i)\right]\geqslant 0 \end{array}\right\} (i=1,2\ldots,N). \end{array}\right\} \tag{8}$$

The resulting λ^* will be the minimax error magnitude. Prefixing λ to the vector of coefficients $\mathbf{a}$, we redefine

$$\mathbf{a}=(\lambda, a_1, a_2, \ldots, a_\nu)$$

and define

$$\mathbf{c}=(f(\xi_1),-f(\xi_1),f(\xi_2),-f(\xi_2),\ldots,f(\xi_N),-f(\xi_N)),$$

$$B=\begin{bmatrix} 1 & 1 & 1 & 1 & \ldots & 1 & 1 \\ \Phi_1(\xi_1) & -\Phi_1(\xi_1) & \Phi_1(\xi_2) & -\Phi_1(\xi_2) & \ldots & \Phi_1(\xi_N) & -\Phi_1(\xi_N) \\ \Phi_2(\xi_1) & -\Phi_2(\xi_1) & \Phi_2(\xi_2) & -\Phi_2(\xi_2) & \ldots & \Phi_2(\xi_N) & -\Phi_2(\xi_N) \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ \Phi_\nu(\xi_1) & -\Phi_\nu(\xi_1) & \Phi_\nu(\xi_2) & -\Phi_\nu(\xi_2) & \ldots & \Phi_\nu(\xi_N) & -\Phi_\nu(\xi_N) \end{bmatrix},$$

$$\mathbf{b} = \begin{Bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix}, \qquad \mathbf{w} = \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_N \\ v_N \end{Bmatrix}.$$

Then the linear program (8) can be written in the standard form:

$$\left.\begin{array}{l} \text{Minimize } \mathbf{ab} \\ \text{subject to } \mathbf{a}B \geqslant \mathbf{c} \\ \qquad\qquad (\mathbf{a} \text{ not sign-restricted}) \end{array}\right\} \tag{9}$$

where the inequality applies to every component of $\mathbf{a}B$ and $\mathbf{c}$.

Because of the (usually) large number $2N$ of inequality constraints $\mathbf{a}B \geqslant \mathbf{c}$ in (9), it is much more efficient to solve the problem in the dual form:

$$\left.\begin{array}{l} \text{Maximize } \mathbf{cw} \\ \text{subject to } B\mathbf{w} = b \\ \qquad\qquad \mathbf{w} \geqslant 0 \end{array}\right\} \tag{10}$$

[see, for example, Hadley (1962) for a discussion of duality theory in linear programming]. In the final optimal basic solution of (10), at most $\nu + 1$ of the $2N$ components of $\mathbf{w}$ are nonzero, and these correspond to points at which the weak inequalities in (7) are equalities, i.e., to extremal error points, with one sign associated with the u_i and the other with the v_i.

The labor involved in solving the linear program (10) by the revised simplex method may be taken as roughly proportional to the number of pivots (though of course a certain amount of labor is required to obtain an initial basic feasible solution). The number of pivots required depends strongly on the number of unknowns $\nu + 1$, but only very weakly on the number of points N (increasing N from 100 to 1000 typically increases the number of pivots required by one or two). Thus, quite large problems can be solved very efficiently by this process.

The authors believe that linear programming is the method of choice in this problem. The usual expositions of the exchange process for determining best approximations on discrete point sets make a strong assumption about the linear independence of the functions $\Phi_i(x)$, namely that every ν— by $-\nu$ submatrix of the matrix

$$\begin{bmatrix} \Phi_1(\xi_1) & \Phi_1(\xi_2) & \cdots & \Phi_1(\xi_N) \\ \Phi_2(\xi_1) & \Phi_2(\xi_2) & \cdots & \Phi_2(\xi_N) \\ \vdots & \vdots & & \vdots \\ \Phi_\nu(\xi_1) & \Phi_\nu(\xi_2) & \cdots & \Phi_\nu(\xi_N) \end{bmatrix}$$

is nonzero—an assumption clearly not satisfied in the spline function case. If one attempts to remove this restriction, which is in fact unnecessary, one is led to a process which is essentially equivalent to linear programming. The relationship between exchange processes and linear programming has been elucidated by Powell [Handscomb (1966), Chapter 8].

3. Computational Methods for the Variable-Joint Case

The previous section gives a subroutine for calculating

$$\lambda^*(\mathbf{u}) = \min_{\mathbf{a}} \max_{\alpha \leqslant x \leqslant \beta} |f(x) - A(x; \mathbf{a}, \mathbf{u})|. \tag{11}$$

We now want to calculate

$$\lambda^{**} = \min_{\mathbf{u}} \lambda^*(\mathbf{u})$$

and the associated joints $\mathbf{u}^*$.

The most obvious procedure is to search in the $(m-1)$-dimensional space for a minimum of the function $\lambda^*(\mathbf{u})$. The great efficiency of the linear programming subroutine for calculating $\lambda^*(\mathbf{u})$ indeed renders such approaches feasible for modest values of m, and successful computations have been reported [Esch and Eastman (1967)]. As might be anticipated, the usual poor performance of steepest descent is observed, and other search processes, such as the Fletcher-Powell variable metric descent process (1963), are much more effective. However, the increase in dimensionality of the space in which one must search as the number of joints is increased and the severe problems of numerical differentiation which arise are serious limitations. Furthermore, it appears that $\lambda^*(\mathbf{u})$ is usually insensitive to small changes in $\mathbf{u}$ near the optimal $\mathbf{u}^*$—that is, small variations in $\mathbf{u}$ about $\mathbf{u}^*$ produce only slight changes in $\lambda^*(\mathbf{u})$. It is desirable therefore to develop an optimization method that is more efficient by virtue of taking advantage of specific characterization properties of the problem.

Let it be required then to optimize both $\mathbf{a}$ and $\mathbf{u}$ in the approximating function (2); the total number of free parameters is

$$\text{no. of parameters} = n + 2m - 1 \tag{12}$$

and we hope for a best approximation which equioscillates (i.e., attains maximum error magnitude with alternating sign) on a full set of $n+2m$ critical points; such an approximation can be identified *a posteriori* as a solution by Schumaker's characterization results [(1967), Theorem 4.2]. For purposes of

constructing an iterative scheme, we relax the continuity requirement to C^{n-2}; that is, we work with approximating functions of the form

$$A(x) = \sum_{i=0}^{n} d_i x^i + \sum_{k=1}^{m-1} \frac{c_k}{n!}(x-u_k)_+^n + \sum_{k=1}^{m-1} \frac{F_k}{(n-1)!}(x-u_k)_+^{n-1} \tag{13}$$

where F_k gives the discontinuity of $A^{(n-1)}(x)$ at the kth joint.

We note that, when the joints u_k are regarded as given, this $A(x)$ contains $n+2m-1$ free parameters (the *same* number as in Eq. (12)), and consequently, a best approximation with $n+2m$ error extrema can be anticipated for any joint locations. Such an approximation $A(x)$ is, of course, not the desired solution because of the discontinuities in its $(n-1)$st derivative. The basic idea of this approach, however, is to devise an algorithm for adjusting the joint locations in a manner that tends to reduce the discontinuities in $A^{(n-1)}(x)$. If an iterative process can be devised which generates a sequence of best C^{n-2} approximations in which these discontinuities approach zero, then the limit will be the desired solution, since it will be C^{n-1} and it will possess the full complement of $n+2m$ error extrema.

With given joints $\mathbf{u}$, let the parameters of this $A(x)$ be chosen, for example by the linear programming method of Section 2 above, so that $A(x)$ is a best C^{n-2} approximation to $f(x)$; we may denote the resulting values of the parameters by putting tildes over them. Thereby are defined $(m-1)$ functions $\tilde{F}_k(u_1, \ldots, u_{m-1})$, $k = 1, \ldots, m-1$, of the $(m-1)$ quantities $u_1, \ldots, u_{m-1}$. Our endeavor is to solve the nonlinear system

$$\tilde{F}(u_1, \ldots, u_{m-1}) = 0, \qquad k = 1, \ldots, m-1. \tag{14}$$

We employ various modifications of Newton's method; that is, given joint locations $\mathbf{u}$, we define new "improved" joint locations by the linearized equations

$$\tilde{F}_k + \sum_{j=1}^{m-1} \delta u_j \left(\frac{\partial \tilde{F}_k}{\partial u_j}\right) = 0, \qquad k = 1, \ldots, m-1, \tag{15}$$

introducing perhaps an under-relaxation parameter α_p in the typical pth iteration of the process to help ensure convergence:

$$u_j^{(p+1)} = u_j^{(p)} + \alpha_p \, \delta u_j. \tag{16}$$

One approach is to calculate the partial derivatives in (15) by numerical differentiation:

$$\left(\frac{\partial \tilde{F}_k}{\partial u_l}\right)_1 \doteq \frac{\tilde{F}_k(u_1, \ldots, u_l + \Delta u_l, \ldots, u_{m-1}) - \tilde{F}_k(u_1, \ldots, u_l, \ldots, u_{m-1})}{\Delta u_l} \tag{17}$$

Each of the two values of $\tilde{F}_k$ in Eq. (17) is calculated by solving (by linear programming) a C^{n-2} best approximation problem with the indicated

joints. Substitution of the resulting values into Eqs. (15) is tantamount to replacing each surface $\tilde{F}_k(u_1, u_2, \ldots, u_{m-1})$ by a secant plane coinciding with the surface at the m points $(u_1, u_2, \ldots, u_{m-1})$, $(u_1 + \Delta u_1, u_2, \ldots, u_{m-1})$, $\ldots$, $(u_1, u_2, \ldots, u_{m-1} + \Delta u_{m-1})$; therefore the method has been called the *secant plane method.* Care must be taken to avoid the familiar pitfalls of numerical differentiation when small Δu_l are employed.

The labor of setting up the linear system (15) on each iteration is substantial, requiring m fixed-joint best approximation calculations (and perhaps more, if special steps are taken to safeguard against numerical differentiation troubles). Therefore it is of interest to look for a less laborious iterative process. First, we neglect the dependence of $\tilde{F}_k$ on all the u_j except u_k; i.e., we neglect all save the diagonal elements of (15). Secondly, we assume that most of the dependence of $\tilde{F}_k$ on u_k is explicit, so that the implicit dependence which results from the fact that coefficients $\tilde{a}_j$ all change as u_k changes can be neglected. The result is a simple approximate formula for $\partial \tilde{F}_k/\partial u_k$ which requires only one fixed-joint best approximation calculation. If the best approximation for joints $\mathbf{u}$ has the representation

$$\tilde{A}(x; \mathbf{a}, \mathbf{u}) = \sum_{i=0}^{n} t_{k,i} x^i \qquad u_{k-1} \leqslant x \leqslant u_k \tag{18}$$

in the typical kth interval between joints, then the above approximations yield

$$\left(\frac{\partial \tilde{F}_k}{\partial u_l}\right)_2 = \delta_{k,l} n! [t_{k+1,n} - t_{k,n}]. \tag{19}$$

The *simplified Newton's method* which results from substitution of (19) into (15) has been found quite successful in practice. Indeed the success of this (and also of the previously described secant plane method) has been such that trial of more sophisticated recently developed methods for nonlinear systems has not seemed necessary. The ultimate convergence of both methods is quadratic. Unlike the search for otpimal $\lambda^*(\mathbf{u})$, the search for the zeros of the $\tilde{F}_k(\mathbf{u})$ in the $(m-1)$-dimensional joint space appears to be well-conditioned; small variations in $\mathbf{u}$ about $\mathbf{u}^*$ tend to result in large changes in the $\tilde{F}_k(\mathbf{u})$.

4. Sample Results and Computational Experience

Optimal joint best spline approximations have been computed for many test cases using the two methods described above. Sample results are given in Tables I and II for the functions $f(x) = e^x$ on $[0, 1]$, and $\sqrt{x}$ on $[0, 1]$. The former was chosen as an example of a well-behaved analytic function, and the latter as an example of a function exhibiting rather difficult properties (note the vertical tangent at $x = 0$). As might be anticipated, in the "well-behaved" case $f(x) = e^x$ increasing the degree n is more advantageous than adding joints,

TABLE I

OPTIMAL-JOINT SPLINE APPROXIMATIONS TO $f(x) = e^x$ ON $[0,1]$[a]

m = Number of Sub-Intervals

n = Degree of Spline	1	2	3	4	5	6	7	8
0	.859141	.429570		.214785				.107393
1	.105933							
2	.008756	.001342 .5429	.000427 .3809 .6951	.00018701 .2934 .5418 .7713	.00009789 .2386 .4440 .6363, .8170	.00005750 .2011 .3762, .5416 .6984, .8475	.00003659 .1737, .3263 .4715, .6100 .7424, .8692	.00002472 .1530, .2882 .4175, .5416 .6607, .7752 .8856
3	.00054478	.00006396 .5322	.00001597 .3833 .6748	.00000572 .2996 .5315 .7507	.00000253 .2458 .4385 .6222, .7979	.00000128 .2085 .3733, .5314 .6836, .8302		
4	.00002716	.00000255 .5258	.00000052 .388 .660	.00000016 .308 .527 .736				
5	.00000113	.00000009 .523	.00000002 .41 .67					
6	.00000004							

[a] The first entry in each box is λ, the minimax error magnitude. Subsequent entries give the optimal joint locations.

TABLE II

OPTIMAL-JOINT SPLINE APPROXIMATIONS TO $f(x) = \sqrt{x}$ ON $[0,1]$[a]

m = Number of Sub-Intervals

n = Degree of Spline

	1	2	3	4
0	.500000	.250000 .2500	.166667 .1111 .4444	.125000 .0625 .2500 .5625
1	.125000	.041667 .1111		
2	.067619	.019158 .0590	.008226 .01088 .1523	.004347 .00305 .04144 .2381
3	.045928	.012227 .0395	.004931 .00645 .1076	
4	.034686	.008943 .0295		
5	.027842	.0070 .023		
6	.023242			
7	.019946			
8	.017465			

[a] The first entry in each box is λ, the minimax error magnitude. Subsequent entries give the optimal joint locations.

whereas in the "difficult" case $\sqrt{x}$ on [0,1] improvement is very slow as n increases, and adding joints, i.e., increasing the number of subintervals m, is more effective in reducing the minimax error magnitude λ.

The initial assumption on which these methods are based, that the best C^{n-1} spline approximation with optimal joint locations has an error which equioscillates on a full complement of $n+2m$ critical points, was observed to hold in all examples considered (which included, in addition to those shown in Tables I and II, optimal joint calculations for various n and m for $f(x)=\operatorname{erf}(5x)$, $1-e^{-15x}$, various nonanalytic functions, $(1+[kx]^2)^{-1}$ for various values of k, etc., all on [0,1]). It even appears that allowing the joints to assume their optimum locations not only adds $m-1$ critical points, but tends to suppress the "non-normal" cases that appear in the fixed-joint theory, the joints being free to move where they are most needed.

TABLE III

APPROXIMATIONS TO e^x ON [0,1]; MESH: .01

Type	Degree	Joint locations	Continuity class	Degrees of freedom	Minimax deviation
Polynomial	2		C^2	3	.008756
	3		C^3	4	.000544
	4		C^4	5	.000027
	5		C^5	6	.000001
Spline	2	.5429	C^1	4	.001342
	3	.5	C^2	5	.000082
	3	.5322	C^2	5	.000064
	4	.5258	C^3	6	.000002
	3	.38, .67	C^2	6	.000016
	3	.30, .53, .75	C^2	7	.000006
Piecewise	3	.5	C^{-1}	8	.000041
	3	.5	C^0	7	.000043
	3	.5	C^1	6	.000067
(Spline)	3	.5	C^2	5	.000082
(Polynomial)	3	—	C^3	4	.000544

The methods used are perfectly applicable to cases where higher order discontinuities are allowed. Only in exceptional cases are these optimal (as in quadratic spline approximation of $f(x)=|x|$ in $[-1,1]$). Ordinarily the situation is similar to that shown in Table III, where the last section shows the relatively small additional improvement that results in allowing a higher order discontinuity in a piecewise cubic approximation to e^x. (In Tables III and IV, C^{-1} indicates the absence of any continuity requirement at the joints.) The optimal joint locations for spline approximations to e^x are relatively evenly

distributed over [0, 1], and a heavy penalty results from bringing joints together to form higher-order discontinuities. Table IV shows a case where this effect is less marked than usual.

TABLE IV

APPROXIMATIONS TO erf(5x) ON [0,1]; MESH: .002

Type	Degree	Joint locations	Continuity class	Degrees of freedom	Minimax deviation
Polynomial	2		C^2	3	.15785
	3		C^3	4	.05279
	4		C^4	5	.01512
	5		C^5	6	.01418
	6		C^6	7	.00762
	7		C^7	8	.00223
Spline	2	.309	C^1	4	.00834
	3	.516	C^2	5	.01187
	4	.1	C^3	6	.01182
	3	.293, .58	C^2	6	.00575
Piecewise	3	.29	C^{-1}	8	.00470
	3	.29	C^0	7	.00561
	3	.29	C^1	6	.00572
(Spline)	3	.516	C^2	5	.01187
(Polynomial)	3	—	C^3	4	.05279

A computational difficulty associated with the discretization occurs in the case of first-degree splines. The process will often terminate with a spurious result in which the error maximum is achieved at the mesh points on either side of a joint. This trouble can be prevented by a program modification which at each step inserts the current joint locations into the discrete point set $Y = \{\xi_1, \xi_2, \ldots \xi_N\}$. (In general, it is wise to include in the set Y any point at which the error curve can have a cusp.)

The authors do not mean to imply that one should go to the trouble of optimizing joint locations in all applications. Frequently it is quite satisfactory to meet accuracy requirements by using a few additional fixed joints at locations chosen in accordance with some *a priori* rule. Attention might also be given to the possibility of experimentally adjusting joint locations by the use of a scope display and light pen.

ACKNOWLEDGMENT

This work was supported by the Digital Computation Division, Systems Engineering Group, U.S. Air Force, Wright-Patterson AFB, Ohio, under Contracts No. AF33(615)-3257 and F33615-67-C-1570.

References

BARRODALE, I. AND YOUNG, A. (1967). A note on numerical procedures for approximation by spline functions. *Computer J.* **9**, 318–320.

ESCH, R. E. AND EASTMAN, W. L. (1967). Computational Methods for Best Approximation. Tech. Rep. SEG-TR-67-30, Aeronautical Systems Division, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio.

ESCH, R. E. AND EASTMAN, W. L. (1968). Computational Methods for Best Approximation and Associated Numerical Analyses. Tech. Rep. ASD-TR-68-37, Aeronautical Systems Division, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio.

FLETCHER, R. AND POWELL, M. J. D. (1963). A rapidly convergent descent method for minimization. *Computer J.* **6**, 163–168.

HADLEY, G. (1962). "Linear Programming." Addison-Wesley, Reading, Massachusetts.

HANDSCOMB, D. C., Ed. (1966). "Methods of Numerical Approximation." Pergamon, Oxford.

LAWSON, C. L. (1963). Segmented Rational Minimax Approximation, Characteristic Properties and Computational Methods. Jet Propulsion Laboratory Tech. Rep. 32-579, Pasadena, California.

LAWSON, C. L. (1964). Characteristic properties of the segmented rational minimax approximation problem. *Numerische Math.* **6**, 293–301.

POWELL, M. J. D. (1966). On Best L_2 Spline Approximations. Atomic Energy Research Establishment Report T.P.264, Harwell, Berkshire, England.

RICE, J. R. (1964). "The Approximation of Functions." Addison-Wesley, Reading, Massachusetts.

RICE, J. R. (1967). Characterization of Chebyshev approximations by splines. *SIAM J. Numerical Anal.* **4**, 557–565.

RIVLIN, T. J. AND CHENEY, E. W. (1966). A comparison of uniform approximations on an interval and a finite subset thereof. *SIAM J. Numerical Anal.* **3**, 311–320.

SCHUMAKER, L. L. (1967a). Uniform Approximation by Tchebycheffian Spline Functions; I. Fixed Knots. Mathematics Research Center Report No. 768, Madison, Wisconsin.

SCHUMAKER, L. L. (1967b). Uniform Approximation by Tchebycheffian Spline Functions; II. Free Knots. Mathematics Research Center Report No. 810, Madison, Wisconsin.

SCHUMAKER, L. L. (1968). On Computing Best Spline Approximations. Mathematics Research Center Report No. 833, Madison, Wisconsin.

SHISHA, O. (1966). The Chebyshev polynomial of best approximation to a given function on an interval. *Math. Computation* **20**, 266–271.