$$\forall_i\forall_j\forall_k[M_{i-a_k,j-b_k} \leftarrow M_{i-a_k,j-b_k} + P_{ij}]. \tag{4}$$

This can be done in parallel for all the points in P by changing the order of the operations, i.e.,

$$\forall_k\forall_i\forall_j[M_{i-a_k,j-b_k} \leftarrow M_{i-a_k,j-b_k} + P_{ij}]. \tag{5}$$

In other words, to each point in M_{ij} we add the value of $P_{i-a_k,j-b_k}$ and iterate this procedure for all the pairs (a_k,b_k) that define the 180° rotation of the given curve. With this procedure, we have to compute each value for the increment index (a_k and b_k) only once for all the points in P_{ij}, which implies that processing time is decreased.

But major gain is achieved by the implementation of the algorithm in a parallel machine [5], [6]. In such a machine, the algorithm can be executed by translations of the P_{ij} matrix and additions of the translated array to the M_{ij} matrix; the total number of these parallel operations is the same as the number of points in the discrete plane X-Y that define the curve to be detected.

Now, after the accumulation of all the traces, the best candidate for A' will correspond to the cell with the maximum value in the matrix M, because through this cell passes the maximum number of traces, as explained in Section II.

The method described is also suitable for the case in which the picture P_{ij}, has different grey levels.

ACKNOWLEDGMENT

The authors gratefully acknowledge the helpful comments of the members of the Pattern Recognition Group, School of Engineering, University of California, Irvine, during the preparation of this paper. Particular thanks go to Prof. J. Sklansky, the Chairman of this group.

REFERENCES

[1] P. V. C. Hough, "Method and means for recognizing complex patterns," U. S. Patent 3 069 654, Dec. 18, 1962.
[2] R. O. Duda and P. E. Hart, "Use of Hough transformation to detect lines and curves in pictures," *Commun. Ass. Comput. Mach.*, vol. 15, Jan. 1972.
[3] A. K. Griffith, "Edge detection in simple scenes using *a priori* information," *IEEE Trans. Comput.*, vol. C-22, pp. 371–381, Apr. 1973.
[4] C. Kimme, D. Ballard, and J. Sklansky, "Finding circles by an array of accumulators," School of Engineering, Univ. of Calif., Irvine, Interim Rep. IP-73-5.
[5] J. Sklansky, "A parallel natural image computer," School of Engineering, Univ. of Calif., Irvine, personal communications, Sept. 1966.
[6] B. Kruse, "A parallel picture processing machine," *IEEE Trans. Comput.*, vol. C-22, pp. 1075–1087, Dec. 1973.

Optimal Piecewise Polynomial L_2 Approximation of Functions of One and Two Variables

THEODOSIOS PAVLIDIS

***Abstract*—The problem of piecewise polynomial L_2 approximation with variable boundaries is considered. Necessary and sufficient conditions for local optima are derived. These suggest simple functional iteration algorithms for locating the boundaries.**

***Index Terms*—L_2 approximation, Newton's method, piecewise polynomial approximation, variable boundary approximation, variable breakpoint approximation.**

Manuscript received November 1, 1973; revised June 10, 1974. This work was supported by the National Science Foundation under Grant GK-36180.

The author is with the Department of Electrical Engineering, Princeton University, Princeton, N. J. 08540.

I. INTRODUCTION

The problem of piecewise polynomial approximation occurs in a number of applications and it has been dealt with in the recent literature. Of particular interest is the case when the "breakpoints" of the approximation are allowed to vary [1]–[4] (See [4] for additional references to the earlier mathematical literature.)

We present here criteria for optimality for the case of the L_2 norm for both functions of one two variables. These criteria depend only on the local errors at the dividing boundaries, and they suggest algorithms for solving the problem of optimal location of breakpoints.

II. THE SINGLE-VARIABLE CASE

Let $f(x)$ be a continuous differentiable function on $[a,b]$ which is to be approximated by polynomials $\{p_i(x)\}_{i=1}^n$ of given degree $m-1$ on intervals $(x_{i-1},x_i]_{i=1}^n$ where

$$a = x_0 < x_1 \cdots < x_{n-1} < x_n = b \tag{1}$$

in order to minimize

$$E = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} [f(x) - p_i(x)]^2\, dx. \tag{2}$$

It is well known that under the above assumptions, we can use differentiation with respect to the breakpoints and polynomial coefficients to define conditions for the optimal solution.

We introduce the following notation:

$$\varphi(t) = \mathrm{col}\,\{1,t,t^2,\cdots,t^{m-1}\} \tag{3a}$$

$$\boldsymbol{a}_i = \mathrm{col}\,\{a_{0i},a_{1i},a_{2i},\cdots,a_{m-1,i}\} \tag{3b}$$

$$\boldsymbol{F}_i = \int_{x_{i-1}}^{x_i} f(t)\varphi(t)\, dt, \qquad x_i > x_{i-1} \tag{3c}$$

$$\boldsymbol{M}_i = \int_{x_{i-1}}^{x_i} \varphi(t)\varphi'(t)\, dt, \qquad x_i > x_{i-1} \tag{3d}$$

$$p_i(t) = \sum_{j=0}^{m-1} a_{ji}t^j = \boldsymbol{a}_i'\varphi(t) \tag{3e}$$

$$e_i(t) = f(t) - \boldsymbol{a}_i'\varphi(t) \tag{3f}$$

$$e_i^{(1)}(t) = \frac{df}{dt} - \boldsymbol{a}_i'\frac{d\varphi}{dt}. \tag{3g}$$

The primed symbols denote the transpose of a vector.

The matrix $\boldsymbol{M}_i$ is usually referred to as the *Gram matrix* and it is symmetric and positive definite [5].

Taking the partial derivatives of E with respect to the coefficients a_{ji} of the polynomials yields the well-known [5], [6] system of linear equations

$$\frac{\partial E}{\partial a_{ji}} = \int_{x_{i-1}}^{x_i} [f(x) - p_i(x)]x^j\, dx = 0, \qquad i = 1,2,\cdots,n, j = 0,1,\cdots,m-1 \tag{4}$$

or in vector form

$$\operatorname{grad}_{\boldsymbol{a}_i} E = \boldsymbol{F}_i - \boldsymbol{M}_i\boldsymbol{a}_i = \boldsymbol{0}, \qquad i = 1,2,\cdots,n \tag{4'}$$

while the partial derivatives with respect to the x_i's result in

$$\frac{\partial E}{\partial x_i} = [f(x_i) - p_i(x_i)]^2 - [f(x_i) - p_{i+1}(x_i)]^2 = 0, \qquad i = 1,2,\cdots,n-1 \tag{5}$$

$$\frac{\partial E}{\partial x_i} = e_i^2(x_i) - e_{i+1}^2(x_i) = 0, \qquad i = 1,2,\cdots,n-1, \tag{5'}$$

i.e., a necessary condition that the breakpoints are optimally located is that the absolute values of the pointwise errors from right and left are equal. This means that at any breakpoint, the approximation is either continuous or symmetric with respect to $f(t)$.

A sufficient condition for a minimum of a function of many variables is that the matrix of the second derivatives be positive definite. In order to avoid taking these derivatives with respect to the coefficients a_{ji} as well as the x_i's, it is necessary to substitute into (5) the expressions for the a_{ji}'s (as functions of the x_i's) obtained by solving (4). The calculations are laborious but straightforward and can be found in the Appendix. The result is

$$\frac{\partial^2 E}{\partial x_i \partial x_{i-1}} = \frac{2B_2 \cdot e_i(x_i) e_i(x_{i-1})}{x_i - x_{i-1}} \tag{6a}$$

$$\frac{\partial^2 E}{\partial x_i^2} = 2e_i(x_i)e_i^{(1)}(x_i) - 2e_{i+1}(x_i)e_{i+1}^{(1)}(x_i) - \frac{2B_1 \cdot e_i^2(x_i)}{x_i - x_{i-1}} - \frac{2B_1 \cdot e_{i+1}^2(x_i)}{x_{i+1} - x_i} \tag{6b}$$

$$\frac{\partial^2 E}{\partial x_i \partial x_{i+1}} = \frac{2B_2 \cdot e_{i+1}(x_i) e_{i+1}(x_{i+1})}{x_{i+1} - x_i} \tag{6c}$$

$$\frac{\partial^2 E}{\partial x_i \partial x_k} = 0, \qquad \text{otherwise} \tag{6d}$$

where B_1 and B_2 are given by

$$B_1 = m^2$$

$$B_2 = (-1)^{m-1}m.$$

Therefore we have the following.

Theorem: A configuration of breakpoints $x_1, x_2, \cdots, x_{n-1}$ minimizes E, as given by (2), if and only if the approximation at any breakpoint is either continuous or symmetric with respect to $f(t)$ and the matrix given by (6) is positive definite.

Verifying the latter condition may not be easy, and therefore we proceed to derive some simpler conditions. Roughly speaking, the first two terms in (6b) reflect the change in

$$e_i^2(x_i) - e_{i+1}^2(x_i)$$

caused directly by a variation in x_i, while the remaining terms in all equations reflect the indirect change due to the modification of the approximating polynomials for the new breakpoints. Intuitively, one expects that these will be less significant. Indeed, all such terms in (6) are proportional to the *ratio of the pointwise errors at the x_i's divided by the length of the intervals*. On the other hand, the first two terms of (6b) depend on the *derivatives of the pointwise errors*. For any reasonable problem, the pointwise errors should be small in comparison to the interval length, and therefore the matrix of the second derivatives has diagonal dominance.

In the case of a continuous solution [i.e., $e_i(x_i) = e_{i+1}$], we have

$$\partial^2 E/\partial x_i^2 \cong 2e_i(x_i)[e_i^{(1)}(x_i) - e_{i+1}^{(1)}(x_i)] \tag{7a}$$

and in the case of a symmetric solution [i.e., $e_i(x_i) = -e_{i+1}(x_i)$],

$$\partial^2 E/\partial x_i^2 \cong 2e_i(x_i)[e_i^{(1)}(x_i) + e_{i+1}^{(1)}(x_i)]. \tag{7b}$$

These expressions also can be written as

$$\partial^2 E/\partial x_i^2 \cong 2e_i(x_i)[p_i^{(1)}(x_i) - p_i^{(1)}(x_i)] \tag{7a$'$}$$

$$\partial^2 E/\partial x_i^2 \cong 2e_i(x_i)[2f^{(1)}(x_i) - p_i^{(1)}(x_i) - p_{i+1}^{(1)}(x_i)]. \tag{7b$'$}$$

We can summarize now.

Corollary 1: A sufficient condition for optimal breakpoint location is that the approximation is continuous, and that the difference of the slope of the approximation to the right minus that of the left has the same sign as the pointwise error there. Furthermore, the absolute value of that difference must be significantly greater than the ratio of the absolute pointwise errors over the lengths of the intervals. Fig. 1(a) shows such a configuration.

Corollary 2: A sufficient condition for optimal breakpoint location is that the approximation is symmetric, and the difference of the slope of $f(t)$ minus the average slope of the approximation has the same sign as the pointwise error. Furthermore, the absolute value of that difference must be significantly greater than the ratio of the absolute pointwise errors over the lengths of the intervals. Fig. 1(b) shows such a configuration.

The term "significant" must be interpreted according to the order of approximation and the number of breakpoints. It should be emphasized that the simplifications leading to Corollaries 1 and 2 are not necessary in actual computation where all the quantities appearing in (6) may be evaluated exactly (see Section IV).

III. THE TWO-VARIABLE CASE

The situation here is considerably more complex because of the need to define the form of the boundaries which are to be considered. A simple case is the following.

Let $f(x,y)$ be a square integrable function over $[a,b]X[c,d]$ which is to be approximated by polynomials $p_1(x,y)$ and $p_2(x,y)$ over regions R_1 and R_2 such that

$$R_1 = \{(x,y) \mid a \le x < g(y), \qquad c \le y \le d\} \tag{8}$$

$$R_2 = \{(x,y) \mid g(y) \le x < b, \qquad c \le y \le d\} \tag{9}$$

where $g(y)$ is a polynomial of degree m in y. It is required to chose $p_1(x,y)$, $p_2(x,y)$, and $g(y)$ in order to minimize

$$E = \int_c^d \int_a^{g(y)} [f(x,y) - p_1(x,y)]^2 \, dxdy + \int_c^d \int_{g(y)}^b [f(x,y) - p_2(x,y)]^2 \, dxdy. \tag{10}$$

Taking the partial derivatives of E with respect to the coefficients of $g(y)$ and setting them equal to zero yields

$$\int_c^d [f(g(y),y) - p_1(g(y),y)]^2 y^r dy = \int_c^d [f(g(y),y) - p_2(g(y),y)]^2 y^r dy, \qquad r = 0,1,\cdots,m, \tag{11}$$

i.e., a necessary condition that the dividing boundary is optimally located is that the integral of the square error along the boundary multiplied by y^r is the same from right and left.

The following is a more general case. Let each region be defined through

$$R_i = \{(x,y) \mid g_i(\boldsymbol{a}_i,x,y) > 0\}, \qquad i = 1,2,\cdots,n \tag{12}$$

where $\boldsymbol{a}_i$ is a parameter vector. Then

$$E = \sum_{i=1}^n \iint_{g_i(\boldsymbol{a}_i,x,y)>0} [f(x,y) - p_i(x,y)]^2 \, dxdy, \tag{13}$$

which can be written as

$$E = \sum_{i=1}^n \int_c^d \left(\sum_{j=1}^{k_i} \int_{g_{ij}(\boldsymbol{a}_i,y)}^{h_{ij}(\boldsymbol{a}_i,y)} [f(x,y) - p_i(x,y)]^2 \, dx \right) dy \tag{14}$$

where k_i denotes the number of segments along the line $y = \text{const}$ of region R_i and h_{ij} and g_{ij} their right and left endpoints. The partial derivative of E with respect to a parameter a_{gr} is

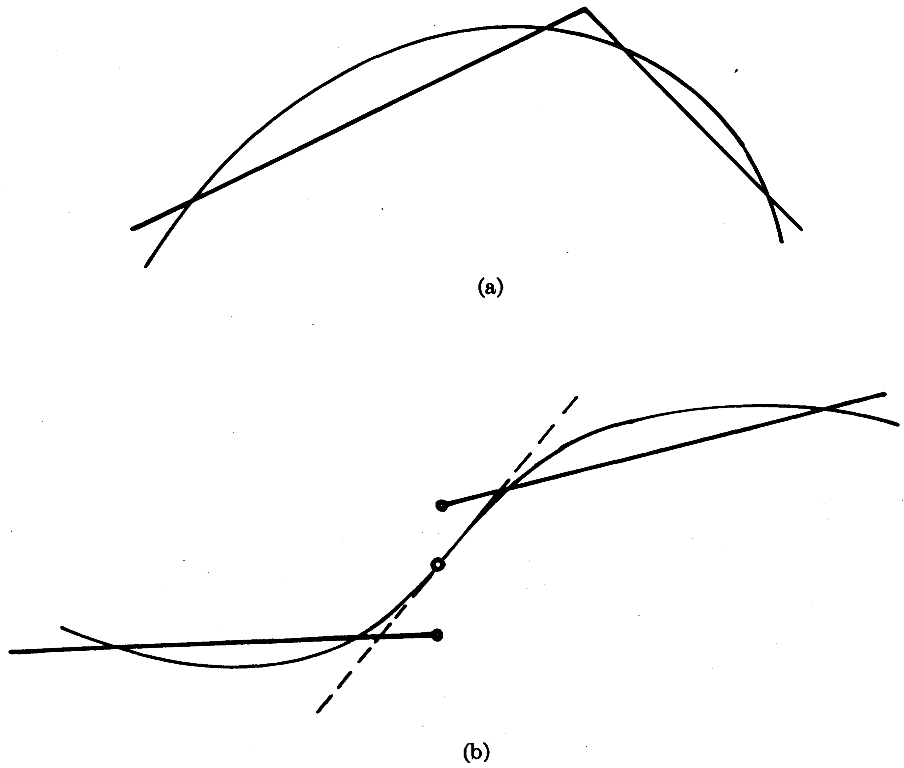

Fig. 1. Form of the polynomial approximation at the breakpoints of an optimal solution.

$$\frac{\partial E}{\partial a_{gr}} = \sum_{i=1}^{n}\left[\int_c^d\left(\sum_{j=1}^{k_i}\left\{[f(h_{ij},y) - p_i(n_{is},y)]^2\frac{\partial h_{ij}}{\partial a_{gr}} - [f(g_{ij},y) - p_1(g_{ij},y)]^2\frac{\partial g_{ij}}{\partial a_{gr}}\right\}\right)dy\right]. \quad (15)$$

This implies again that integrals along the boundaries of the same form as (11) express necessary conditions for optimality.

Sufficient conditions can be expressed in terms of integrals of the derivatives along the boundaries.

The need to specify the form of the regions makes it impossible to derive a solution for the general case. As a matter of fact, this problem is ill-defined unless a proper choice of boundary description is made [7].

IV. ALGORITHMS

Equation (5) suggests that candidates for the optimal solution can be found as zeros of the vector-valued function $V(x)$:

$$V_i(x) = e_{i+1}(x_i)^2 - e_i(x_i)^2, \qquad i = 1,2,\cdots,n-1. \quad (16)$$

Let J denote the Jacobian matrix $(\partial V_i/\partial x_j)$. This is obviously equal to the tridiagonal matrix of the second derivatives of E given by (6). Under the rather general assumptions of Corollaries 1 and 2, it has diagonal dominance, and therefore the application of Newton's method in vector form is particularly appropriate [8]. Thus the following iterative scheme is applicable:

$$x^{k+1} = x^k - J^{-1}V(x^k). \quad (17)$$

If the terms with the factors B_i in (6) are sufficiently small, then (17) can be simplified into

$$x_i^{k+1} = x_i^k + \frac{e_{i+1}(x_i)^2 - e_i(x_i)^2}{2[e_{i+1}(x_i)e_{i+1}^{(1)}(x_i) - e_i(x_i)e_i^{(1)}(x_i)]}, \qquad i = 1,2,\cdots,n-1. \quad (18)$$

It is also possible to use a first-order iteration method similar to that used for the case of the L_∞ norm [4], [9], namely, the scheme

$$x_i^{k+1} = x_i^k + c[e_{i+1}(x_i)^2 - e_i(x_i)^2], \qquad i = 1,2,\cdots,n-1 \quad (19)$$

where

$$c < \inf_k\left\{\frac{i}{\max_i |\partial V_i/\partial x_i|}\right\} \quad (20)$$

with the partial derivatives evaluated at x_i^k. The assumption that $f(x)$ is differentiable together with (7) guarantees that c is finite. However, this scheme will be, in general, slower than those given by (17) or (18). Cases where

$$\partial V_i/\partial x_i = 0$$

can be excluded since this implies that a local change of the breakpoint will have no effect. Such points can be ignored at intermediate iteration steps.

Similar schemes can be devised for the case of functions of two variables where coefficients of the boundary curves are varied instead of breakpoints.

A split-and-merge algorithm [10] can be used to obtain a starting point for these techniques since they converge only in a neighborhood of a solution. They can also be combined with such an algorithm in order to accelerate convergence.

In the past, the optimization of the location of the breakpoints has been done by using descent techniques [3], [4], [10]. It has been observed that the locations obtained in this way satisfy the conditions of Corollaries 1 and 2 [10, Figs. 7 and 9]. The use of Newton's method is expected to result in faster convergence.

CONCLUSIONS

The simple form of the optimality conditions suggests solutions which are simpler than minimizing the L_2 norm via steepest descent schemes [3], [4]. Furthermore, the same schemes are extendable to the case of functions of two variables.

APPENDIX

We will compute here the second derivatives of E after the expressions for the optimal coefficients of the polynomial have been substituted in (2). We note the well-known results [5], [6]

$$a_i = M_i^{-1}F_i \tag{A.1}$$

and

$$E = \int_a^b f(t)^2\,dt - \sum_{i=1}^{m-1} F_i' a_i. \tag{A.2}$$

A direct differentiation of (3c) and (3d) yields

$$\frac{\partial F_i}{\partial x_k} = \begin{cases} f(x_i)\varphi(x_i), & \text{if } k = i \\ -f(x_{i-1})\varphi(x_{i-1}), & \text{if } k = 1 - 1 \\ 0, & \text{otherwise} \end{cases} \tag{A.3}$$

$$\frac{\partial M_i}{\partial x_k} = \begin{cases} \varphi(x_i)\varphi'(x_i), & \text{if } k = i \\ -\varphi(x_{i-1})\varphi'(x_{i-1}), & \text{if } k = i - 1 \\ 0, & \text{otherwise.} \end{cases} \tag{A.4}$$

Differentiating (A.1) gives

$$\frac{\partial a_i}{\partial x_k} = -M_i^{-1}\frac{\partial M_i}{\partial x_k}M_i^{-1}F_i + M_i^{-1}\frac{\partial E_i}{\partial x_k}. \tag{A.5}$$

Substitution of (A.3) and (A.4) into (A.5) followed by a substitution from (3e) results in

$$\frac{\partial a_i}{\partial x_i} = M_i^{-1}\varphi(x_i)e_i(x_i) \tag{A.6a}$$

$$\frac{\partial a_i}{\partial x_{i-1}} = -M_i^{-1}\varphi(x_{i-1})e_i(x_{i-1}). \tag{A.6b}$$

We proceed now with the differentiation of (A.9) while using (A.3)–(A.6) and (3f):

$$\begin{aligned} \frac{\partial E}{\partial x_j} &= -\sum_{i=1}^{n}\left[\frac{\partial F_i'}{\partial x_j}a_i + F_i'\frac{\partial a_i}{\partial x_j}\right] \\ &= -f(x_j)\varphi'(x_j)a_j - F_j'M_j^{-1}\varphi(x_j)e_j(x_j) \\ &\quad + f(x_j)\varphi'(x_j)a_{j+1} + F_{j+1}'M_{j+1}^{-1}\varphi(x_j)e_{j+1}(x_j) \\ &= 2f(x_j)\varphi'(x_j)(a_{j+1} - a_j) + [\varphi'(x_j)a_j]^2 - [\varphi'(x_j)a_{j+1}]^2 \\ &= [\varphi'(x_j)(a_{j+1} - a_j)]\cdot[2f(x_j) - \varphi'(x_j)a_j - \varphi'(x_j)a_{j+1}] \\ &= [e_j(x_j) - e_j(x_j)][e_{j+1}(x_j) + e_j(x_j)] \\ &= e_j^2(x_j) - e_{j+1}^2(x_j). \end{aligned} \tag{A.7}$$

As expected, this is the same as (5).
Furthermore, we have

$$\frac{\partial^2 E}{\partial x_j \partial x_k} = 2e_j(x_j)\frac{\partial e_j(x_j)}{\partial x_k} - 2e_{j+1}(x_j)\frac{\partial e_{j+1}(x_j)}{\partial x_k}. \tag{A.8}$$

Using (3f), we obtain

$$\frac{\partial e_j(x_j)}{\partial x_k} = \frac{\partial f(x_j)}{\partial x_k} - a_j'\frac{\partial \varphi(x_j)}{\partial x_k} - \varphi'(x_j)\frac{\partial a_j}{\partial x_k}. \tag{A.9}$$

The first two terms are zero unless $j = k$ and then they equal $e_j^{(1)}(x_j)$. The last term is zero unless $k = j$ or $k = j - 1$.
Using (A.6) we find

$$\frac{\partial e_j(x_j)}{\partial x_j} = e_j^{(1)}(x_j) - \varphi'(x_j)M_j^{-1}\varphi(x_j)e_j(x_j) \tag{A.10a}$$

$$\frac{\partial e_j(x_j)}{\partial x_{j-1}} = \varphi'(x_j)M_j^{-1}\varphi(x_{j-1})e_j(x_{j-1}). \tag{A.10b}$$

Similarly, we find

$$\frac{\partial e_{j+1}(x_j)}{\partial x_{j+1}} = -\varphi'(x_j)M_{j+1}^{-1}\varphi(x_{j+1})e_{j+1}(x_{j+1}) \tag{A.10c}$$

$$\frac{\partial e_{j+1}(x_j)}{\partial x_j} = e_{j+1}^{(1)}(x_j) + \varphi'(x_j)M_{j+1}^{-1}\varphi(x_j)e_{j+1}(x_j). \tag{A.10d}$$

The above expressions can be simplified on the basis of the following lemma.

Lemma: If $\varphi(t)$ is the vector defined by (3a) and M the corresponding Gram matrix for the interval (q,r), then

$$Q(p,q,r,s) \equiv \varphi'(p)M^{-1}\varphi(s) = \frac{B}{r-q} \tag{A.11}$$

if p and s take the values r or q. The constant B does not depend on r or q.

Proof: First we observe that the quadratic form Q is invariant under a basis transformation. Indeed, if

$$\Psi(t) = P\varphi(t)$$

we have

$$\begin{aligned} &\Psi'(p)\left\{\int_q^r \Psi(t)\Psi'(t)\,dt\right\}^{-1}\Psi(s) \\ &= \varphi'(p)P'\left\{P\left[\int_q^r \varphi(t)\varphi'(t)\,dt\right]P'\right\}^{-1}P\varphi(s) \\ &= \varphi'(p)\left\{\int_q^r \varphi(t)\varphi'(t)\,dt\right\}\varphi(s). \end{aligned}$$

P can be the matrix of the Gram–Schmidt orthonormalizing process [5], [6], and therefore Q can be evaluated for a set of orthonormal functions equivalent to those of $\varphi(x)$. In that case, the Gram matrix equals the identity matrix and

$$Q(p,q,r,s) = \Psi'(p)\Psi(s). \tag{A.12}$$

Next we note that if $\Psi_i(x)$ is a member of a set of orthonormal polynomials on the interval $[q,r]$, then

$$\Psi_i(x) = \left(\frac{2}{q-r}\right)^{1/2}\Psi_i\left(\frac{2x-(q+r)}{q-r}\right)$$

where $\Psi_i(x)$ denotes the corresponding polynomial on the interval $[-1,1]$ [6, p. 201].
Therefore

$$\Psi_i(q) = \left(\frac{2}{q-r}\right)^{1/2}\Psi_i(1) \tag{A.13a}$$

$$\Psi_i(r) = \left(\frac{2}{q-r}\right)^{1/2}\Psi_i(-1). \tag{A.13b}$$

It is easy to verify that $|\Psi_i(1)| = |\Psi_i(-1)|$, and therefore

$$Q(q,q,r,q) = Q(r,q,r,r) = \frac{2}{q-r}\sum_{i=1}^{m}\Psi_i^2(1) \tag{A.14a}$$

$$Q(q,q,r,r) = Q(r,q,r,q) = \frac{2}{q-r}\sum_{i=1}^{m}\Psi_i(1)\Psi_i(-1). \tag{A.14b}$$

This completes the proof of the lemma since the sums in (14) do not depend on q or r. For brevity, we will use the notations

$$B_1 = 2\sum_{i=1}^{m} \Psi_i^2(1) \tag{A.15a}$$

$$B_2 = 2\sum_{i=1}^{m} \Psi_i(1)\Psi_i(-1). \tag{A.15b}$$

A simple calculation can show that

$$B_1 = m^2 \tag{A.16a}$$

$$B_2 = (-1)^{m-1}m. \tag{A.16b}$$

Using the above lemma in (10), we obtain

$$\frac{\partial e_j(x_j)}{\partial x_j} = e_j^{(1)}(s_j) - \frac{B_1 e_j(x_j)}{x_j - x_{j-1}} \tag{A.17a}$$

$$\frac{\partial e_j(x_j)}{\partial x_{j-1}} = \frac{B_2 e_j(x_{j-1})}{x_j - x_{j-1}} \tag{A.17b}$$

$$\frac{\partial e_{j+1}(x_j)}{\partial x_{j+1}} = \frac{B_2 e_{j+1}(x_{j+1})}{x_{j+1} - x_j} \tag{A.17c}$$

$$\frac{\partial e_{j+1}(x_j)}{\partial x_j} = e_{j+1}^{(1)}(x_j) + \frac{B_1 e_{j+1}(x_j)}{x_{j+1} - x_j}. \tag{A.17d}$$

Substituting (A.17) into (A.8), we obtain (6).

REFERENCES

[1] J. R. Rice, *The Approximation of Functions*, vol. II. Reading, Mass.: Addison-Wesley, 1969, p. 123.
[2] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973, pp. 335–339.
[3] A. Cantoni, "Optimal curve fitting with piecewise linear functions," *IEEE Trans. Comput.*, vol. C-20, pp. 59–67, Jan. 1971.
[4] T. Pavlidis, "Waveform segmentation through functional approximation," *IEEE Trans. Comput.*, vol. C-22, pp. 689–697, July 1973.
[5] P. J. Davis, *Interpolation and Approximation*. New York: Blaisdell, 1963.
[6] E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*. New York: Wiley, 1966.
[7] G. Meinardus, *Approximation of Functions: Theory and Numerical Methods*. Berlin: Springer-Verlag, 1967.
[8] J. F. Traub, *Iterative Methods for the Solution of Equations*. Englewood Cliffs, N. J.: Prentice-Hall, 1964, ch. 11.
[9] T. Pavlidis and A. P. Maika, "Uniform piecewise polynomial approximation with variable joints," *J. Approximation Theory*, vol. 12, pp. 61–69, Sept. 1974.
[10] T. Pavlidis and S. L. Horowitz, "Segmentation of plane curves," *IEEE Trans. Comput.*, vol. C-23, pp. 860–870, Aug. 1974.

The Constrained-Input Problem

FRANK MARKHAM BROWN

***Abstract*—Given a combinational output function f and an input constraint $\phi = 0$, there is a set $G(f, \phi)$ of output functions equivalent to f with respect to ϕ. A function belongs to $G(f, \phi)$, that is, provided its evaluations agree with those of f for all argument combinations satisfying the constraint $\phi = 0$. We define the *constrained-input problem* as that of generating $G(f, \phi)$, given f and ϕ. A general solution for this problem is developed. Applications to the "don't-care" problem and to translator synthesis are discussed.**

***Index Terms*—Boolean algebra, Boolean equations, functional decomposition, input constraints.**

Manuscript received August 4, 1973; revised July 9, 1974. A preliminary version of this paper was presented at the 1973 IEEE Region 3 Conference, Louisville, Ky.
The author is with the Department of Electrical Engineering, University of Kentucky, Lexington, Ky. 40506.

I. INTRODUCTION

We shall consider the problem of designing a logic circuit whose inputs $x_1,x_2,\cdots,x_n$ are constrained by a system of simultaneously asserted logical relations. These relations are assumed to be of the *universal* type (i.e., equations or inclusions, expressed by the affirmative copulas $=$ or $\leq$, respectively). Relations of the *particular* type, expressed by the negative copulas $\neq$ or $\nless$, will not be considered. The Boolean identity $[a \leq b] \equiv [a\bar{b} = 0]$ allows us to transform inclusions into equations; hence, we shall assume that the relations among the input variables are expressed by a system of Boolean equations of the form

$$\begin{aligned} \alpha_1(x_1,\cdots,x_n) &= \beta_1(x_1,\cdots,x_n) \\ \alpha_2(x_1,\cdots,x_n) &= \beta_2(x_1,\cdots,x_n) \\ &\vdots \\ \alpha_k(x_1,\cdots,x_n) &= \beta_k(x_1,\cdots,x_n). \end{aligned} \tag{1}$$

Applying the standard theory of Boolean equations [3], [4], [7], [10], the system (1) is equivalent to the single equation

$$\phi(x_1,x_2,\cdots,x_n) = 0 \tag{2}$$

where the function ϕ is defined by the formula

$$\phi = \sum_{i=1}^{k} (\alpha_i \oplus \beta_i). \tag{3}$$

Let us call (2) the *input constraint* and ϕ the *constraint function.*

The logic-circuit under discussion is to realize the switching function

$$z = f(x_1,x_2,\cdots,x_n). \tag{4}$$

Given f and ϕ, there is a set $G(f,\phi)$ of switching functions, exactly one of which is identical to f, such that if $g \in G$, then $g(\mathbf{x}) = f(\mathbf{x})$ for all $\mathbf{x}$ satisfying the input constraint (2). The logic-circuit will produce satisfactory z-signals if (and only if) it is wired to realize a function in the set $G(f,\phi)$; to find the best function (by any criterion) it is necessary to be able to generate the set $G(f,\phi)$. We therefore define the *constrained-input problem* as follows. Given two n-variable switching functions f and ϕ, construct the set

$$G(f,\phi) = \{g \mid \phi(\mathbf{x}) = 0 \Rightarrow g(\mathbf{x}) = f(\mathbf{x})\} \tag{5}$$

of n-variable switching functions, where $\mathbf{x} = (x_1,x_2,\cdots,x_n)$ is a vector of switching variables, i.e., elements of the two-element Boolean algebra.

If the input constraint reduces to an identity (i.e., if the inputs are independent), then $G(f,\phi)$ has exactly one member, namely, the function f. If the input constraint is nontrivial, then $G(f,\phi)$ contains f, together with functions distinct from f.

II. CONSTRUCTION OF $G(f,\phi)$

The constrained-input problem can be phrased as follows. Find a general solution for g in the relation

$$\phi = 0 \Rightarrow g = f. \tag{6}$$

The relation (6) is equivalent to the Boolean equation

$$\bar{\phi}\cdot(\bar{g}f + g\bar{f}) = 0 \tag{7}$$

which is equivalent in turn to the constraint

$$\bar{\phi}f \leq g \leq \phi + f. \tag{8}$$

The constraint (8) provides a general solution for g; however, the prescription

$$g = \bar{\phi}f + p\phi, \tag{9}$$

expressing the same information, is sometimes more convenient.