MATHEMATICS OF COMPUTATION, VOLUME 26, NUMBER 119, JULY, 1972

Generalized Recursive Multivariate Interpolation*

By Earl H. McKinney

Abstract. A generalized recursive interpolation technique for a set of linear functionals over a set of general univariate basis functions has been previously developed. This paper extends these results to restricted multivariate interpolation over a set of general multivariate basis functions. When the data array is a suitable configuration (e.g., an n-dimensional simplex), minimal degree multivariate interpolating polynomials are produced by this recursive interpolation scheme. By using product rules, recursive univariate interpolation applied to each variable singly produces multivariate interpolating polynomials (not of minimal degree) when the data are arranged in a hyper-rectangular array. By proper ordering of points in a data array, multivariate polynomial interpolation is accomplished over other arrays such as diamonds and truncated diamonds in two dimensions and their counterparts in n dimensions.

1. Introduction. The recursive procedure developed will permit the determination of a multivariate interpolating function which interpolates a given data set over a set of multivariate basis functions. Walker [1], in an unpublished thesis, extended previous work by Thacher [2] and produced a generalized recursive interpolation technique for a given set of linear functionals (not restricted to function values alone) over a set of general univariate basis functions. Later, Newbery [3] published results similar to those of Walker with the restriction that the basis functions are specialized to univariate algebraic and trigonometric polynomial basis functions. The author will generalize some of Walker's results (and hence Newbery's) to restricted multivariate interpolation; i.e., the linear functional data will be function values only.

A review of Walker's results for recursive univariate interpolation is included in Section 2. In Section 3, recursive univariate interpolation is generalized by considering repeated recursive univariate interpolation over hyper-rectangular arrays. This method is illustrated by a general 3×3 array. In Section 4, a theorem is given establishing generalized recursive multivariate interpolation along with a corollary specializing the theorem to recursive multivariate polynomial interpolation over a particular configuration of data points in a specified order. This corollary determines the unique interpolating polynomial of total degree d in n variables. Comparisons with other methods as well as different configurations of data points are discussed in Section 5. Error analysis and tests for convergence to a predetermined accuracy appear in Section 6. The advantages of the recursive interpolation scheme are treated in Section 7 and a suggestion for further investigation appears in Section 8.

Received August 5, 1971.

AMS 1969 *subject classifications.* Primary 6520; Secondary 4110, 4115.

Key words and phrases. Recursive multivariate interpolation, repeated recursive univariate interpolation, hyper-rectangular array, basis functions.

* Work done in part under the auspices of the U. S. Atomic Energy Commission. Research supported by N. S. F. PACE Program Grant GZ-0024 and Ball State University.

The following notation will be adopted to represent points in Euclidean n-space R_n. A generic point in R_n is denoted by

$$p^{(n)} = (x^{(1)}, x^{(2)}, x^{(3)}, \cdots, x^{(n)}).$$

If S is a nonempty subset of R_n, then a specific point $p_N^{(n)} \in S$ is denoted by

$$p_N^{(n)} = (x_{N_1}^{(1)}, x_{N_2}^{(2)}, \cdots, x_{N_n}^{(n)}).$$

When interpolating for functions of several variables using polynomial basis functions, the form of the interpolating polynomial will appear as

$$P(p^{(n)}) = \sum a_N \prod_{k=1}^{n} \prod_{i=0}^{N_{k-1}} (x^{(k)} - x_i^{(k)}),$$

where the summation is over a prescribed data set $\{p_j^{(n)}\} \in E_n$, $j = 0, 1, \cdots, N$.

2. Univariate Interpolation. A brief review of the results of Walker and Thacher for generalized recursive univariate interpolation will provide the background needed to proceed to the multivariate case.

An inductive algorithm was developed by Walker [1] which is suitable for any distribution of base points for which linearly independent basis functions can be constructed. The following theorem establishes the algorithm for constructing the sequence of interpolating functions.

THEOREM 1. *Given a set of base points* $\{x_j\}$, *and function values* $f(x_j)$ $(j = 0, 1, 2, \cdots, m)$, *there exists a sequence of interpolating functions* $\{R_j(x)\}$ *defined recursively by*

$$\begin{aligned} &R_j(x) = R_{j-1}(x) + a_j\varphi_j(x), \qquad j = 1, 2, 3, \cdots, m, \\ &R_0(x) = f(x_0), \quad \text{where } a_j = \frac{f(x_j) - R_{j-1}(x_j)}{\varphi_j(x_j)}, \end{aligned} \tag{1}$$

such that

$$R_k(x_j) = f(x_j), \qquad j = 0, 1, 2, \cdots, k;\ k = 0, 1, 2, \cdots, m, \tag{2}$$

where the $\varphi_j(x)$ *is any set of linearly independent basis functions which satisfy*

$$\begin{aligned} &\varphi_j(x_i) = 0, \qquad i = 0, 1, 2, 3, \cdots, (j-1), \\ &\varphi_j(x_j) \neq 0. \end{aligned} \tag{3}$$

Proof. The proof is by induction. By (1), $R_0(x) = f(x_0)$. Assume that $R_{k-1}(x_j) = f(x_j)$ for $j < k$. Then by (1), $R_k(x_j) = R_{k-1}(x_j) + a_k\varphi_k(x_j)$ and, since $\varphi_k(x_j) = 0$ for $j < k$ by (3), we have $R_k(x_j) = R_{k-1}(x_j) = f(x_j)$ for $j < k$. For $j = k$, $R_k(x_k) = R_{k-1}(x_k) + a_k\varphi_k(x_k)$. But by (1), $a_k = [f(x_k) - R_{k-1}(x_k)]/\varphi_k(x_k)$. Hence, $R_k(x_k) = R_{k-1}(x_k) + \{[f(x_k) - R_{k-1}(x_k)]/\varphi_k(x_k)\}\varphi_k(x_k) = f(x_k)$ and the induction is complete.

COROLLARY 1. *A set of polynomial basis functions for the unique mth degree interpolating polynomial for the data set* $\{x_j\}$ *and function values* $f(x_j)$ $(j = 0, 1, 2, 3, \cdots, m)$ *is given by*

$$\varphi_0(x) = 1, \qquad \varphi_j(x) = (x - x_{j-1})\varphi_{j-1}(x) \qquad (j = 1, 2, 3, \cdots, m). \tag{4}$$

This set of basis functions satisfies (3), *and hence* (1) *produces the required mth degree interpolating polynomial.*

A possible set of basis functions for generalized polynomial interpolation is given by

$$\varphi_0(x) = 1, \qquad \varphi_i(x) = [g_i(x) - g_i(x_{j-1})]\varphi_{j-1}(x),$$

where the $g_i(x)$ are arbitrary functions except that $g_i(x_i) \neq g_i(x_{j-1})$ for $i \geqq j$.

Newbery observed in [3] that the interpolating polynomial obtained from (1) recursively using basis functions (4) is the same as Newton's divided difference formula when the symmetry property of divided differences is taken into account. This algorithm produces the interpolated value explicitly for a given x and also permits retention of the calculated constants a_i for interpolation at other values of x.

3. Repeated Recursive Univariate Interpolation (Hyper-Rectangular Arrays). An acceptable procedure for multivariate interpolation over hyper-rectangular arrays in n dimensions is obtained using repeated univariate interpolation formulas as suggested in Milne et al. [4] and Steffensen [6]. Except for Steffensen's notation for divided differences (or the square bracket notation of other authors), there appears to be a lack of suitable notation for representing repeated univariate rules when performing multivariate operations. In what follows, the symbol I will represent interpolation with respect to a single variable over a suitable data set. (It should be noted that I could be replaced by any suitable operator for generating repeated univariate rules such as numerical differentiation or numerical integration, for example.)

Consider a given hyper-rectangular array in n dimensions denoted by the set

$$\{x_{i_1}^{(1)}, x_{i_2}^{(2)}, x_{i_3}^{(3)}, \cdots, x_{i_n}^{(n)}\} \tag{5}$$

with the corresponding set of function values

$$\{f(x_{i_1}^{(1)}, x_{i_2}^{(2)}, x_{i_3}^{(3)}, \cdots, x_{i_n}^{(n)})\}, \tag{6}$$

i_j nonegative integers with $0 \leqq i_j \leqq N_j$, $j = 0, 1, 2, 3, \cdots, n$. Let (6) be denoted by nests of sets as follows:

$$\{\cdots\{\{f(x_{i_1}^{(1)}, x_{i_2}^{(2)}, \cdots, x_{i_n}^{(n)})\}_{i_1=0}^{N_1}\}_{i_2=0}^{N_2}\cdots\}_{i_n=0}^{N_n}. \tag{7}$$

For example, for a 3×3 two-dimensional data set, (7) is represented as

$$\{\{f(x_i, y_j)\}_{i=0}^{2}\}_{j=0}^{2} = \{\{f(x_0, y_0), f(x_1, y_0), f(x_2, y_0)\}, \{f(x_0, y_1), f(x_1, y_1), f(x_2, y_1)\},$$
$$\{f(x_0, y_2), f(x_1, y_2), f(x_2, y_2)\}\}.$$

In order to interpolate $f(p^{(n)})$ over the data set (5) with function values given by (6) using repeated univariate rules, one must interpolate with respect to each variable, in turn, over the appropriate data set (i.e., with respect to the variables singly). Let us denote interpolation with respect to the variable $x^{(j)}$ by the operator symbol $\mathrm{I}_{i_j=0}^{N_j}(x^{(j)})$ with evaluation at $\bar{x}^{(j)}$. Then

$$\mathop{\mathrm{I}}_{i_j=0}^{N_j}(x^{(j)})f = \{\cdots\{\{\cdots\{P_{N_j}(x_{i_1}^{(1)}, x_{i_2}^{(2)}, \cdots, x_{i_{j-1}}^{(j-1)}, \bar{x}^{(j)},$$
$$x_{i_{j+1}}^{(j+1)}, \cdots, x_{i_n}^{(n)})\}_{i_1=0}^{N_1}\cdots\}_{i_{j-1}=0}^{N_{j-1}}\}_{i_{j+1}=0}^{N_{j+1}}\cdots\}_{i_n=0}^{N_n} \tag{8}$$

which is a collection of $(n-1)$ nested sets (having started with n nested sets of function values) of interpolates. These interpolates are denoted by

$$P_{N_j}(x_{i_1}^{(1)}, x_{i_2}^{(2)}, \cdots, x_{i_{j-1}}^{(j-1)}, \bar{x}^{(j)}, x_{i_{j+1}}^{(j+1)}, \cdots, x_{i_n}^{(n)})$$

to indicate polynomials of degree N_j in the case for polynomial interpolation. (Of course, rational functions, trigonometric functions, etc. could be used depending upon the choice of basis functions for interpolation.)

Repeated univariate interpolation requires interpolation to be performed upon the set of interpolates obtained in (8). Continuing this process, one obtains

$$f(p^{(n)}) \approx \prod_{j=1}^{n} \mathop{\mathrm{I}}_{i_j=0}^{N_j} (x^{(j)})f = P_{N_1 N_2 \cdots N_n}(\bar{x}^{(1)}, \bar{x}^{(2)}, \bar{x}^{(3)}, \cdots, \bar{x}^{(n)})$$

which, in the case for polynomial interpolation, produces an interpolating polynomial with terms of total degree as high as $N_1 + N_2 + N_3 + \cdots + N_n$ but not all lower degree terms.

As an example, consider the data set $\{(x_i, y_j, z_k)\}$ with corresponding function values $\{f(x_i, y_j, z_k)\}$, $i, j, k = 0, 1, 2$. We interpolate at $(\bar{x}, \bar{y}, \bar{z})$ as follows:

First, with respect to x at $\bar{x}$,

$$\mathop{\mathrm{I}}_{i=0}^{2} (x)f = \{\{P_2(\bar{x}, y_j, z_k)\}_{j=0}^{2}\}_{k=0}^{2},$$

then interpolate on this set of interpolates, with respect to y at $\bar{y}$,

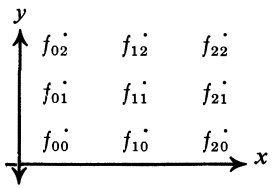
$$\mathop{\mathrm{I}}_{j=0}^{2} (y) \mathop{\mathrm{I}}_{i=0}^{2} (x)f = \{P_{22}(\bar{x}, \bar{y}, z_k)\}_{k=0}^{2},$$

and lastly interpolate on these interpolates, with respect to z at $\bar{z}$,

$$\mathop{\mathrm{I}}_{k=0}^{2} (z) \mathop{\mathrm{I}}_{j=0}^{2} (y) \mathop{\mathrm{I}}_{i=0}^{2} (x)f = P_{222}(\bar{x}, \bar{y}, \bar{z}) \approx f(\bar{x}, \bar{y}, \bar{z}).$$

With this notation established, we can use the recursive univariate interpolation procedure of Theorem 1, repeated over each variable in turn, to accomplish interpolation recursively for multivariate functions over hyper-rectangular arrays of data points.

Example. Use repeated recursive univariate interpolation over the data set $\{x_i, y_j\}$ with corresponding function values $\{f_{ij}\}$, $i, j = 0, 1, 2$.



By choosing the φ functions as appropriate polynomials, Theorem 1 allows us to generate interpolating polynomials recursively with respect to x at $\bar{x}$ over each y_j $(j = 0, 1, 2)$

$$R_0(\bar{x}, y_j) = f_{0j}, \qquad \text{where } a_{ij} = \frac{f_{ij} - R_{i-1}(x_i, y_j)}{\prod_{k=0}^{i-1}(x_i - x_k)},$$

$$R_1(\bar{x}, y_j) = R_0(\bar{x}, y_j) + a_{1j}\varphi_1(\bar{x}),$$

$$R_2(\bar{x}, y_j) = R_1(\bar{x}, y_j) + a_{2j}\varphi_2(\bar{x}), \qquad \varphi_i(\bar{x}) = \prod_{k=0}^{i-1}(\bar{x} - x_k).$$

Hence, $\mathrm{I}_{i=0}^{2}(x)f = \{R_2(\bar{x}, y_j)\}_{j=0}^{2}$ is a set of three interpolating polynomials each of degree 2 in $\bar{x}$. We next interpolate with respect to y upon these interpolates

$$R_{20}(\bar{x}, \bar{y}) = R_2(\bar{x}, y_0), \qquad \text{where } a_j(\bar{x}) = \frac{R_2(\bar{x}, y_j) - R_{2,j-1}(\bar{x}, y_j)}{\prod_{k=0}^{j-1}(y_j - y_k)},$$

$$R_{21}(\bar{x}, \bar{y}) = R_{20}(\bar{x}, \bar{y}) + a_1\varphi_1(\bar{y}),$$

$$R_{22}(\bar{x}, \bar{y}) = R_{21}(\bar{x}, \bar{y}) + a_2\varphi_2(\bar{y}), \qquad \varphi_j(\bar{y}) = \prod_{k=0}^{j-1}(\bar{y} - y_k).$$

The interpolating polynomial for the given data set is therefore given by

$$\mathrm{I}_{j=0}^{2}(y)\,\mathrm{I}_{i=0}^{2}(x)f = R_{22}(\bar{x}, \bar{y}). \tag{9}$$

It is interesting to note the form of (9), observing that the $\varphi_i(\bar{x})$ and $\varphi_j(\bar{y})$ are polynomials of degree i and j, respectively, and the $a_j(\bar{x})$ are polynomials of degree 2 in $\bar{x}$, while the a_{ij} are constants.

$$R_{22}(\bar{x}, \bar{y}) = f_{00} + \sum_{i=1}^{2} a_{i0}\varphi_i(\bar{x}) + \sum_{j=1}^{2} a_j\varphi_j(\bar{y}).$$

$R_{22}(\bar{x}, \bar{y})$ is therefore a polynomial in the variables $\bar{x}$, $\bar{y}$ with terms f_{00}, $\bar{x}$, $\bar{x}^2$, $\bar{y}$, $\bar{x}\bar{y}$, $\bar{x}^2\bar{y}$, $\bar{y}^2$, $\bar{x}\bar{y}^2$, $\bar{x}^2\bar{y}^2$.

It should be observed that repeated recursive univariate interpolation by the method described above produces Newton's divided difference formula for polynomial interpolation of functions over hyper-rectangular arrays in n-space which appears in both [4] and [6].

4. Recursive Multivariate Interpolation. The recursive scheme described in Section 2 can be generalized to multivariate interpolation for data points in other than hyper-rectangular arrays.

Using the notation for points in E_n defined in Section 1, the following generalization of Theorem 1 results.

THEOREM 2. *Given a set of base points* $p\{_N^{(n)}\}$, $N = 0, 1, 2, \cdots, T$, *and function values* $f(p_N^{(n)})$, *there exists a sequence of interpolating functions* $\{R_N(p^{(n)})\}$ *defined by*

$$R_N(p^{(n)}) = R_{N-1}(p^{(n)}) + a_N\varphi_N(p^{(n)}), \qquad N = 1, 2, \cdots, T,$$
$$R_0(p^{(n)}) = f(p_0^{(n)}), \qquad a_N = \frac{f(p_N^{(n)}) - R_{N-1}(p_N^{(n)})}{\varphi_N(p_N^{(n)})} \tag{10}$$

such that

$$R_N(p_j^{(n)}) = f(p_j^{(n)}), \qquad j = 1, 2, 3, \cdots, N, \tag{11}$$

where $\varphi_N(p^{(n)})$ *is any set of linearly independent basis functions which satisfy*

$$\varphi_N(p_j^{(n)}) = 0, \qquad j = 0, 1, 2, 3, \cdots, (N-1), \tag{12}$$
$$\varphi_N(p_N^{(n)}) \neq 0.$$

Again, a simple inductive proof, similar to that of Theorem 1, establishes that the $R_N(p^{(n)})$, defined recursively by (10) and (12), satisfy (11).

A particular configuration, as well as an ordering, imposed upon the set of data points $p_N^{(n)}$, will permit the selection of polynomial basis functions $\varphi_N(p^{(n)})$ which determines the unique interpolating polynomial of minimal degree satisfying the given data.

In Milne et al. [4, p. 54], $(d+1)(d+2)/2$ data points arranged in a two dimensional trigangular array, with equal spacing of data points in each direction, determines the unique minimal degree interpolating polynomial in two variables including all terms up through total degree d. Milne et al. [4] show that Newton's advancing difference formula for a function of two variables may then be used to determine such an interpolating polynomial. Equal spacing of data points is not required. A Newton divided difference formula may be used if the subscripts on the coordinates of the data points form the specified triangular array.

A choice of polynomial basis functions to determine the unique minimal dth degree interpolating polynomial for a function of n variables with $\binom{n+d}{d}$ data points will now be displayed. We consider as given the $\binom{n+d}{d}$ data points $p_N^{(n)}$ arranged on an n-dimensional rectangular grid such that

$$0 \leqq N < \binom{n+d}{d}, \qquad N_i \geqq 0 \text{ (integers)}, \qquad 0 \leqq \sum_{i=1}^{n} N_i \leqq d. \tag{13}$$

One ordering imposed upon the points $p_N^{(n)}$ satisfying (13) which permits a recursive construction of the minimal degree interpolating polynomials is obtained as follows. Consider the ordering to be imposed upon the subscripts $N_1, N_2, \cdots, N_n$ of the point $p_N^{(n)} = (x_{N_1}^{(1)}, x_{N_2}^{(2)}, \cdots, x_{N_n}^{(n)})$.

Definition 1. ${}_NS_k = \sum_{i=1}^{k} N_i$ is the kth partial sum of the coordinates of $N = (N_1, N_2, \cdots, N_n)$.

Definition 2. $N < M$ if, and only if,

(14)
(a) $${}_NS_n < {}_MS_n \quad \text{or}$$
(b) $${}_NS_m < {}_MS_m \text{ and } {}_NS_i = {}_MS_i \text{ for some integer } m \text{ such that } 0 < m < n \text{ and all integers } i \text{ such that } m < i \leqq n.$$

The ordering imposed by (14) produces a one-to-one mapping of the n-tuples $(N_1, N_2, \cdots, N_n)$ onto the nonnegative integers by

$$N = \sum_{i=1}^{n} \binom{{}_NS_i + j - 1}{j}. \tag{15}$$

The existence of such mappings is well known; for instance, see [5]. The derivation of (15) and the establishment of the one-to-one mapping are tedious but straightforward. (The author will provide a constructive proof of this particular one-to-one mapping upon request.) It should be pointed out that a complete knowledge of the one-to-one

mapping represented by (14) and (15) is not essential to ensuing developments. Fig. 1 indicates the mapping for $n = 2, 3$ established by (15).

To describe the ordering represented by (14), we assert that the n-tuple $(0, 0, 0, \cdots, 0)$ is mapped onto zero. If we denote the successor of $N = (N_1, N_2, N_3, \cdots, N_n)$ by $N' = (N_1', N_2', N_3', \cdots, N_n')$, then (14) imposes the following relations between the N_i and N_i'.

If $N_2 \neq 0$,

$$N' = (N_1 + 1, N_2 - 1, N_3, N_4, \cdots, N_n).$$

If $N_i = 0$, $i = 2, 3, 4, \cdots, k$, ($k = n$ included)

$$N' = (0, 0, 0, \cdots, 0, N_1 + 1, N_{k+1} - 1, N_{k+2}, \cdots, N_n).$$

For example, for $n = 5$, the successor of (1, 3, 2, 0, 3) is (2, 2, 2, 0, 3) and the successor of (2, 0, 0, 0, 3) is (0, 0, 0, 3, 2).

The following corollary establishes a choice of basis functions $\varphi_N(p^{(n)})$ for unique minimal degree polynomial interpolation.

COROLLARY. *A set of polynomial basis functions for the unique interpolating polynomial of total degree d for* $\binom{n+d}{d}$ *data points* $p_N^{(n)} = (x_{N_1}^{(1)}, x_{N_2}^{(2)}, \cdots, x_{N_n}^{(n)})$ *satisfying* (13) *and* (14) *with function values* $f(p_N^{(n)})$ *is given by*

$$\varphi_N(p^{(n)}) = \prod_{k=1}^{n} \prod_{i=0}^{N_k - 1} (x^{(k)} - x_i^{(k)}). \tag{16}$$

(*By the usual* $\prod$ *convention for* $N_k = 0$, *the corresponding factor is* 1.)

Proof. We need to show that conditions (12) are satisfied by the $\varphi_N(p^{(n)})$ given by (16).

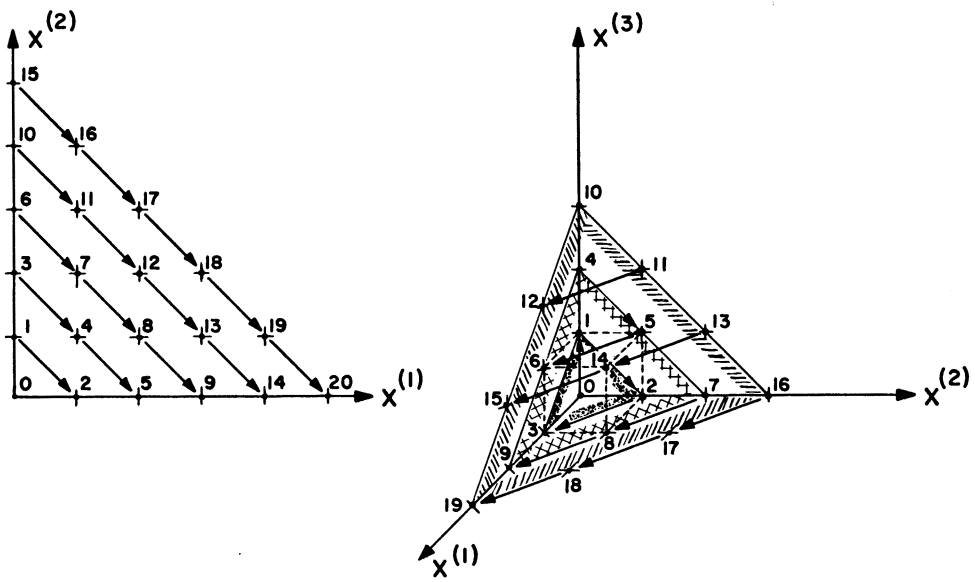


FIGURE 1

Let N^* be any nonnegative integer such that N^* is mapped onto $(N_1^*, N_2^*, N_3^*, \cdots, N_n^*)$. Furthermore, assume $N^* < N$. Then

$$\varphi_N(p_{N^*}^{(n)}) = \prod_{k=1}^{n} \prod_{i=0}^{N_k-1} (x_{N_k^*}^{(k)} - x_i^{(k)}). \tag{17}$$

By (14)(a), $\sum_{k=1}^{n} N_i^* < \sum_{k=1}^{n} N_i$ implies $N_k^* < N_k$ for some k $(1 \leqq k \leqq n)$. By (14)(b), ${}_{N^*}S_m < {}_NS_m$ and ${}_{N^*}S_{m+1} = {}_NS_{m+1}$ for some m implies that $\sum_{i=1}^{m} N_i^* < \sum_{i=1}^{m} N_i$. Since N_i^* and N_i are nonnegative integers, there exists a $k \leqq m$ such that $N_k^* < N_k$. Hence, since $0 \leqq i \leqq N_k - 1$ for all k in (17), then i takes on the value N_k^* and $\varphi_N(p_{N^*}^{(n)}) = 0$ for all $N^* < N$.

That $\varphi_N(p_N^{(n)}) \neq 0$ is obvious from direct substitution of $p_N^{(n)}$ into (16). Therefore, it has been shown that the $\varphi_N(p^{(n)})$ are a proper set of polynomial basis functions satisfying (12).

An ordering other than that chosen in Definition 2 may be selected with the proof of the Corollary remaining essentially unaltered.

A set of generalized polynomial basis functions $\varphi_N(p^{(n)})$ may be chosen in the form

$$\varphi_N(p^{(n)}) = \prod_{k=1}^{n} \prod_{i=0}^{N_k-1} [g_{ik}(x^{(k)}) - g_{ik}(x^{(k)})], \tag{18}$$

where the $g_{ik}(x^{(k)})$ are arbitrary functions except that $g_{ik}(x_j^{(k)}) \neq g_{ik}(x_i^{(k)})$ for $j > i$. The use of (18) in place of (16) in the above corollary does not significantly alter the proof.

5. Configurations and Comparisons. It is important to note that the recursive multivariate polynomial interpolation procedure of Section 4 imposes configuration restrictions upon the subscripts of the data points and not upon the data points themselves. In terms of flexibility in applications, this is roughly equivalent to the difference between using divided differences and ordinary differences.

Consider the case where interpolation is required near the boundary of a table, or some other restrictive boundary, upon the determination of function values for interpolation. We restrict the discussion to two dimensions for ease in representing configurations of data points. For example, the interpolating polynomial of total degree 3, obtained by using the recursive procedure of Section 4, has the form

$$\begin{aligned} P(x, y) = {} & a_{00} + a_{10}(x - x_0) + a_{01}(y - y_0) \\ & + a_{20}(x - x_0)(x - x_1) + a_{11}(x - x_0)(y - y_0) + a_{02}(y - y_0)(y - y_1) \\ & + a_{30}(x - x_0)(x - x_1)(x - x_2) + a_{21}(x - x_0)(x - x_1)(y - y_0) \\ & + a_{12}(x - x_0)(y - y_0)(y - y_1) + a_{03}(y - y_0)(y - y_1)(y - y_2). \end{aligned}$$

In all cases which follow the ordered pairs (i, j), refer to the subscripts on the data point (x_i, y_j) with the order of appearance of the subscripts ij on a_{ij} indicating the order in which the terms, and hence data points, are taken in the recursive scheme. The data points could be arranged as indicated in Table 1 to perform interpolation along boundaries.

For interpolation in other corners or along other sides a reflection or rotation of the configurations given can be used. Additional data points for higher-degree polynomial

interpolation would be distributed by extension of the pattern described in each case.

TABLE 1

					(0 , 3)		
					(0 , 2)	(1 , 2)	
				(2 , 1)	(0 , 1)	(1 , 1)	
				(2 , 0)	(0 , 0)	(1 , 0)	(3 , 0)
(0 , 3)							
(0 , 2)	(1 , 2)						
(0 , 1)	(1 , 1)	(2 , 1)					
(0 , 0)	(1 , 0)	(2 , 0)	(3 , 0)				

For interpolation away from the boundaries of the table for a point (x, y) within the rectangle (x_0, y_0), (x_1, y_0), (x_0, y_1), (x_1, y_1) the following scheme could be used.

TABLE 2. *Central Table Interpolation*

	(0 , 3)		
(2 , 1)	(0 , 1)	(1 , 1)	
(2 , 0)	(0 , 0)	(1 , 0)	(3 , 0)
	(0 , 2)	(1 , 2)	

(Note that the i and j coordinates alternate about (0, 0) in increasing order of each coordinate.) Of course, other assignment schemes are possible and each scheme has an analogous form in higher dimensions. The data point configurations described above are combinations of Newton and Gauss forward and backward interpolation formulas. For example, the central table interpolation scheme above is a Gauss-Gauss interpolation formula (i.e., it is a Gauss interpolation formula with respect to each variable).

Milne et al. [4, Chapter IV] derives the interpolation formulas of Newton, Stirling, and Everett for functions of two variables with equally spaced data points. Special arrays of points are exhibited in the forms of diamonds, squares, and truncated diamonds referred to as 5-point, 9-point, 13-point, 21-point, and 25-point formulas. All of these formulas can be obtained by either repeated recursive univariate interpolation for the square arrays (9-point and 25-point formulas) or by recursive multivariate interpolation by omitting certain terms from total degree formulas.

It is obvious that the square-array formulas of Milne et al. can be handled by repeated recursive univariate interpolation. A scheme for alternating rows and columns about a central square or point, could be arranged in order to have the higher-degree terms contribute relatively small amounts to the total.

As an example of recursive multivariate interpolation applied to the 13-point diamond formula of Milne et al. [4, pp. 56–57], we have the following configuration of data points (Table 3).

This formula is obtained by recursive multivariate interpolation by setting the coefficients a_{31} and a_{13} equal to zero in the recursive scheme for constructing the interpolating polynomial of total degree 4 from the given data. Such a polynomial is

TABLE 3. 13-*Point Diamond Array*—(3, 1) *and* (1, 3) *Omitted*

		(0 , 3)	(1 , 3)	
	(2 , 1)	(0 , 1)	(1 , 1)	(3 , 1)
(4 , 0)	(2 , 0)	(0 , 0)	(1 , 0)	(3 , 0)
	(2 , 2)	(0 , 2)	(1 , 2)	
		(0 , 4)		

of the form

$$P(x, y) = \sum_{m=0}^{4} \sum_{k=0}^{m} a_{m-k,k} \prod_{i=1}^{m-k} (x - x_{i-1}) \prod_{j=1}^{k} (y - y_{j-1}). \tag{19}$$

Hence, the recursive interpolation routine for the polynomial of total degree 4 can be used by the simple expedient of assigning $a_{31} = a_{13} = 0$.

Similar modifications of the routine for recursive multivariate interpolation of predetermined total degree will permit interpolation over a wide range of configurations including the remaining ones discussed by Milne et al.

6. Error Analysis and Convergence Criteria. Since the recursive repeated univariate interpolation procedure for hyper-rectangular arrays developed in Section 3 and the recursive multivariate interpolation procedure developed in Section 4 are analogous to Newton's divided difference formula (in more than one variable), an ideal reference for error analysis is Steffensen [6]. Other forms of the error term for polynomial interpolation in several variables are given by Kincaid [7] and Sard [8]. Kincaid's results are also discussed in Milne et al. [4, pp. 78–80]. Kincaid's error terms do not require any special configuration of points and include the case where the interpolating polynomial need not contain all terms of total degree d in all variables. Sard's error terms are valid for linear functionals in general and are in integral form.

In practical applications of interpolation, one generally does not have the necessary derivatives available in order to analyze the error terms. The alternative then is to assume convergence of the recursive scheme and compare successive iterates or groups of successive iterates and terminate the interpolating process when a predetermined accuracy has been reached. Of course, it is possible for the recursive scheme to converge to a result which satisfies some practical convergence criterion but not provide the prescribed accuracy.

Since convergence criteria for repeated recursive univariate interpolation over hyper-rectangular arrays usually involves comparison of results upon inclusion of additional rows and/or columns of data points in the data set, these results will not be discussed.

In the case of recursive multivariate interpolation, five different convergence criteria were used by the author on a representative collection of functions of two variables with the accuracy of the results compared with the true value in each case. Notationally, we shall use $R_N(p^{(n)}) = R_N(x, y)$ for two variable representation of the interpolating function with $N = (N_1, N_2)$. The criteria used are as follows:

(I) $|R_{N+1}(x, y) - R_N(x, y)| < \epsilon$.

(II) $\sum |R_N(x, y) - R_{N-1}(x, y)|/d < \epsilon$ where the summation is over all $N \ni N_1 + N_2 = d$, $N_1 \neq 0$.

(III) $|R_N(x, y) - R_{N-1}(x, y)| < \epsilon/(d + 1)$ for all $N \ni N_1 + N_2 = d$, $N_1 \neq 0$.

(IV) $|R_N(x, y) - R_{N-(d+1)}(x, y)| < \epsilon$ for $N \ni N_1 = 0$, $N_2 = d$ only.

(V) $|R_N(x, y) - R_{N-(d+1)}(x, y)| < \epsilon$ and $|R_{N-1}(x, y) - R_{N-(d+1)}(x, y)| < \epsilon$ for all $N \ni N_1 + N_2 = d$, $N_1 \neq 0$.

(I) requires that the absolute value of the term added to the interpolating function at the $(N + 1)$st stage be less than ϵ in absolute value.

(II) requires that the average of the absolute values of all terms of degree d in the interpolating function be less than ϵ.

(III) requires that each term of degree d contribute less than $\epsilon/(d + 1)$ to the interpolating function. If $N_1 \neq 0$, only d differences are considered. ($\epsilon/(d + 1)$ is used since there are $d + 1$ terms of total degree d.)

(IV) requires that the absolute value of the difference between the interpolating functions of total degree d and $d - 1$ be less than ϵ.

(V) requires that both the absolute value of the difference between the interpolating function through terms including $x^{N_1}y^{N_2-1}$ and $x^{N_1}y^{N_2}$, respectively, and $x^{N_1-1}y^{N_2}$ and $x^{N_1}y^{N_2}$, respectively, to be less than ϵ for all $N \ni N_1 + N_2 = d$ ($N_1 \neq 0$, $N_2 \neq 0$).

Three functions were used as test functions,

$$f = \exp(x + y),$$

$$g = 1/(x + 10)^2 + 1/(xy + 4) + 1/(y - 6)^2,$$

$$h = 1/(x^2 - y^2 - 2) + 1/(x - y + 2.5).$$

Each function was used under the criteria (I)–(V) above with tolerance $\epsilon = .5 \times 10^{-4}$, $.5 \times 10^{-5}$, $.5 \times 10^{-6}$, $.5 \times 10^{-7}$. The data set was a subset of $(-1, 1) \times (-1, 1)$ with equally spaced points at intervals of length $2/d$ ($d > 1$) in both dimensions where d is the total degree of polynomial used at each stage (d increasing until the respective convergence criteria are met). The order in which the data points are to be taken is indicated in Table 2. After the convergence criteria (I)–(V) were met in each case the interpolated value was compared with the true value at $x = .1$, $y = .075$.

Criterion (I) gave the required accuracy only for function f at $\epsilon = .5 \times 10^{-7}$. Criteria (II)–(V) gave the required accuracy for functions f and g for all tolerances indicated. Criteria (II)–(V) gave the required accuracy for function h only at $\epsilon = .5 \times 10^{-4}$.

Function h has singularities near the data set and hence one would expect to have difficulties obtaining a high degree of accuracy in this case.

7. Advantages of Recursive Multivariate Interpolation. The greatest single advantage of recursive multivariate interpolation (including repeated univariate recursive interpolation over hyper-rectangular arrays) is embodied in the adjective recursive. For digital computer programming, a recursive scheme is usually superior, provided other difficulties are not encountered such as the introduction of an unnecessarily large number of calculations which may cause round-off error to become significant. The example of interpolating for the value of the elliptic integral $F(\phi, \theta)$

at $\phi = 49.6°$ and $\theta = 59.4°$, using the 12-point truncated diamond and the 13-point diamond data sets, was programmed. Milne et al. [4, pp. 62–64] constructs Lagrange-type formulas and interpolates for this elliptic integral over the specified 12-point and 13-point data sets. Each data set requires a different Lagrange-type formula and hence must be programmed separately. By recursive multivariate interpolation, the same program is used with $a_{40} = a_{21} = a_{03} = 0$ in (19) for the 12-point formula and $a_{31} = a_{13} = 0$ in (19) for the 13-point formula. The results of these interpolations agreed with those of Milne et al. to the same number of digits of accuracy.

A further advantage of this method is the ability to provide for the convergence of the iterates to a predetermined tolerance. The Newton, Stirling, Bessell, Everett, and Lagrange-type discussed by Milne et al. [4] or Steffensen [6] are not basically recursive and hence do not easily lend themselves to comparisons of differences of successive results with a predetermined tolerance.

Finally, except for the Lagrange-type formula, the formulas mentioned above, although capable of representation in the most general divided difference form, do not usually so appear, undoubtedly due to the complexity of the notation. Recursive multivariate interpolation does not suffer this disadvantage. In general, the equal mesh-size case does not provide any advantage, notational or otherwise, over the unequal mesh-size case.

Remark. It should be pointed out that the repeated recursive univariate scheme of Section 3 could be used to construct unique total degree d interpolating functions (polynomials) also. To do so, one would include in the set of data points for each variable in turn only those points which contribute appropriate terms for the total degree d interpolating polynomial desired. For example, using the set of data points $\{\{x_i, y_j\}_{i=0}^{d-j}\}_{j=0}^{d}$ produces the array (for $d = 3$)

$$
\begin{array}{cccc}
\cdot & & & \\
\cdot & \cdot & & \\
\cdot & \cdot & \cdot & \\
\cdot & \cdot & \cdot & \cdot
\end{array}
$$

An alternate plan would be to use the method as described in Section 3, setting $a_{ij} = 0$ for $i + j > d$ in the process of calculating the coefficients leading to the interpolating polynomial given by (9). The proper triangular array is thus obtained from the square array previously assumed.

Conversely, the recursive multivariate scheme of Section 4 could be used to perform interpolation over hyper-rectangular arrays of dimension $m_1 \times m_2$ by the simple expedient of setting $a_N = 0$ in (10) if $N_1 > m_1$ or $N_2 > m_2$ where $N = (N_1, N_2)$.

8. Further Investigations. Certain types of combinations of functional and derivative data at selected points in the data set permit Hermite-type interpolation by the recursive multivariate interpolation method. It can be seen that in the case where all points of the data set approach $p_0^{(n)}$, for instance, the recursive multivariate interpolation method produces the Taylor series expansion about $p_0^{(n)}$. Further investigation may provide a method whereby recursive multivariate interpolation will produce a method for interpolating with more general Hermite-type data over n-dimensional data sets.

Department of Mathematics
Ball State University
Muncie, Indiana 47306

1. P. L. WALKER, *Generalized Recursive Interpolation*, Thesis, University of Kansas, Lawrence, Kan., 1964. (Unpublished.)
2. H. C. THACHER, JR., *Inductive Interpolation Algorithm*, 1963. (Unpublished.)
3. A. C. R. NEWBERY, "Interpolation by algebraic and trigonometric polynomials," *Math. Comp.*, v. 20, 1966, pp. 597–599. MR **34** #3752.
4. W. E. MILNE, W. ARNTZEN, N. REYNOLDS & J. WHEELOCK, *Mathematics for Digital Computers*. Vol. I. *Multivariate Interpolation*, WADC Technical Report 57–556, 1958.
5. M. DAVIS, *Computability and Unsolvability*, McGraw-Hill Series in Information Processing and Computers, McGraw-Hill, New York, 1958, pp. 43–46. MR **23** #A1525.
6. J. F. STEFFENSEN, *Interpolation*, 2nd ed., Chelsea, New York, 1950. MR **12,** 164.
7. W. M. KINCAID, "Note on the error in interpolation of a function of two independent variables," *Ann. Math. Statist.*, v. 19, 1948, pp. 85–88. MR **9,** 470; MR **10,** 55.
8. A. SARD, "Remainders: Functions of several variables," *Acta Math.*, v. 84, 1951, pp. 319–346. MR **12,** 680.
9. H. C. THACHER, JR., "Derivation of interpolation formulas in several independent variables," *Ann. New York Acad. Sci.*, v. 86, 1960, pp. 758–775. MR **22** #7243.