MATHEMATICS OF COMPUTATION
VOLUME 50, NUMBER 181
JANUARY 1988, PAGES 189-196

A Note on Piecewise Linear and Multilinear Table Interpolation in Many Dimensions

By Alan Weiser and Sergio E. Zarantonello*

Abstract. This note is concerned with N-dimensional rectangular table interpolation, where N is relatively large (4 to 10). Two interpolants are considered: a piecewise multilinear generalization of piecewise bilinear interpolation on rectangles, and a piecewise linear generalization of piecewise linear interpolation on triangles. We show that the two interpolants have similar approximation properties, but the piecewise linear interpolant is much cheaper to evaluate.

1. Introduction. This note is concerned with N-dimensional rectangular table interpolation, where N is relatively large (4 to 10). Two interpolants are considered: a piecewise multilinear generalization of piecewise bilinear interpolation on rectangles, and a piecewise linear generalization of piecewise linear interpolation on triangles. Both interpolants are second-order accurate, continuous, and monotone, and the gradients of both interpolants can be evaluated with minor additional effort. However, the piecewise linear interpolant is much cheaper to evaluate than the piecewise multilinear interpolant: For the piecewise linear interpolant the dominant computational task is to sort N numbers, and for the piecewise multilinear interpolant the dominant computational task is to perform 2^N multiplies.

Table interpolation in N dimensions, $N > 3$, can be useful in situations where the alternative is to solve many more N-dimensional linear or nonlinear systems of equations than there are entries in the table. However, the storage required by strictly rectangular tables grows quickly with N: A typical table in practice might have ten entries in two or three crucial dimensions, and two or three entries in the other dimensions.

The two-dimensional versions of both interpolants to be considered are discussed in standard numerical analysis texts, e.g. [3]. The piecewise multilinear interpolant is a straightforward extension of piecewise bilinear interpolation on rectangles. The piecewise linear interpolant is essentially first-degree multivariate B-spline interpolation [4], [6] on the Kuhn triangulation of the unit N-cube, e.g. [1], [7]: These are standard tools in multivariate approximation theory and simplex methods for finding fixed points and solutions to nonlinear equations. Vectorization and parallelization issues related to these interpolants are complex, and will not be discussed here. See [5] for a discussion of some of these issues when $N = 1$ and $N = 2$.

Received September 8, 1986; revised March 30, 1987.

1980 *Mathematics Subject Classification* (1985 *Revision*). Primary 65D15; Secondary 65D05, 65D20.

* *Current address of second author*: Amdahl Corporation, 1250 East Arques Avenue, P.O. Box 3470, Sunnyvale, California 94088.

An N-dimensional rectangular table interpolant is an approximation $F(x_1,\dots,x_N)$ to a function $f(x_1,\dots,x_N)$ which is computed from table values of f

$$\{f(X_{i_1},X_{i_2},\dots,X_{i_N})\colon i_1=1,\dots,n_1,\dots,i_N=1,\dots,n_N\}.$$

The gradient of the interpolant is

$$\nabla F(x_1,\dots,x_N)=(F^{x_1},\dots,F^{x_N})^T=\left(\frac{\partial F}{\partial x_1},\dots,\frac{\partial F}{\partial x_N}\right)^T.$$

Both interpolants to be considered first find appropriate table intervals

$$(X_{I_1},X_{I_1+1}),\dots,(X_{I_N},X_{I_N+1})$$

such that

$$X_{I_1}\le x_1\le X_{I_1+1},\dots,X_{I_N}\le x_N\le X_{I_N}.$$

This can be done, for example, by the subroutine INTERV [2]. Scaling each interval to $(0,1)$ reduces the problem to that of finding an interpolant $G(y_1,\dots,y_N)$ which approximates $g(y_1,\dots,y_N)$ in the unit N-cube

$$\Omega=\{(y_1,\dots,y_N)\colon 0\le y_1\le 1,\dots,0\le y_N\le 1\}$$

using the table values

$$\{g(j_1,\dots,j_N)\colon j_1=0 \text{ or } 1,\dots,j_N=0 \text{ or } 1\},$$

where

$$y_i=\frac{x_i-X_{I_i}}{X_{I_i+1}-X_{I_i}} \quad\text{and}\quad g(j_1,\dots,j_N)=f(X_{I_1+j_1},\dots,X_{I_N+j_N}).$$

The two interpolants will now be described.

2. A Piecewise Multilinear Interpolant. The piecewise multilinear interpolant F_M can be computed recursively:

$$\begin{aligned}F_M&=G(y_1,\dots,y_N)\\&=G(y_1,\dots,y_{N-1},0)+y_NG^{y_N}(y_1,\dots,y_N)\end{aligned}$$

where

$$G^{y_N}(y_1,\dots,y_N)=G(y_1,\dots,y_{N-1},1)-G(y_1,\dots,y_{N-1},0).$$

In general,

$$\begin{aligned}&G(y_1,\dots,y_i,j_{i+1},\dots,j_N)\\&\quad=G(y_1,\dots,y_{i-1},0,j_{i+1},\dots,j_N)+y_iG^{y_i}(y_1,\dots,y_i,j_{i+1},\dots,j_N),\end{aligned}$$

where

$$\begin{aligned}&G^{y_i}(y_1,\dots,y_i,j_{i+1},\dots,j_N)\\&\quad=G(y_1,\dots,y_{i-1},1,j_{i+1},\dots,j_N)-G(y_1,\dots,y_{i-1},0,j_{i+1},\dots,j_N)\end{aligned}$$

and

$$G(j_1,\dots,j_N)=g(j_1,\dots,j_N).$$

For example, when $N=3$, letting $y_1=x$, $y_2=y$, and $y_3=z$,

$$G_M=G(x,y,z)=G(x,y,0)+z(G(x,y,1)-G(x,y,0)),$$

where

$$G(x,y,0) = G(x,0,0) + y(G(x,1,0) - G(x,0,0)),$$
$$G(x,y,1) = G(x,0,1) + y(G(x,1,1) - G(x,0,1)),$$

and where

$$G(x,0,0) = G(0,0,0) + x(G(1,0,0) - G(0,0,0)),$$
$$G(x,1,0) = G(0,1,0) + x(G(1,1,0) - G(0,1,0)),$$
$$G(x,0,1) = G(0,0,1) + x(G(1,0,1) - G(0,0,1)),$$
$$G(x,1,1) = G(0,1,1) + x(G(1,1,1) - G(0,1,1)).$$

The gradient ∇F_M can be built up along with F_M by differentiating the recursion for F_M:

$$F_M^{x_i} = G^{y_i}(y_1,\ldots,y_N)/(X_{I_i+1} - X_{I_i}),$$

where for $i < N$,

$$G^{y_i}(y_1,\ldots,y_N) = G^{y_i}(y_1,\ldots,y_{N-1},0) + y_N G^{y_i y_N}(y_1,\ldots,y_N),$$

with

$$G^{y_i y_N}(y_1,\ldots,y_N) = G^{y_i}(y_1,\ldots,y_{N-1},1) - G^{y_i}(y_1,\ldots,y_{N-1},0).$$

In general, for $k < i$,

$$G^{y_k}(y_1,\ldots,y_i,j_{i+1},\ldots,j_N)$$
$$= G^{y_k}(y_1,\ldots,y_{i-1},0,j_{i+1},\ldots,j_N) + y_i G^{y_k y_i}(y_1,\ldots,y_i,j_{i+1},\ldots,j_N),$$

where

$$G^{y_k y_i}(y_1,\ldots,y_i,j_{i+1},\ldots,j_N)$$
$$= G^{y_k}(y_1,\ldots,y_{i-1},1,j_{i+1},\ldots,j_N) - G^{y_k}(y_1,\ldots,y_{i-1},0,j_{i+1},\ldots,j_N).$$

These quantities can all be assembled in a single array of length 2^N, as illustrated in Figure 1 for $N = 3$. The quantities used to compute $G_{xyz} = G(x,y,z)$ are circled.

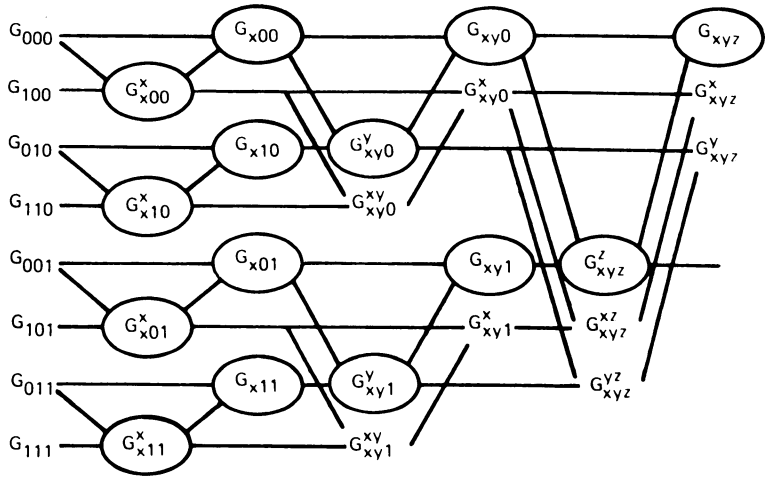

FIGURE 1. *Construction of* $G_M, \nabla G_M$

Computations are performed column-by-column, left-to-right, so that differencing and then Taylor expansion are performed for each successive direction. It takes $2^N - 1$ multiplies to compute F_M and $2^{N+1} - 2$ multiplies to compute both F_M and ∇F_M.

The pointwise error bound for the piecewise multilinear interpolant is straightforward [3], [8].

THEOREM 2.1. *For every $z = (y_1, \ldots, y_N)^T$ in the unit N-cube Ω,*

$$|(g - G_M)(z)| \le z^T(e - z)\|g\|/2,$$

where $e = (1, \ldots, 1)^T$ and the seminorm $\|g\|$ is defined by

$$\|g\| = \sup_{\substack{i=1,\ldots,N \\ y \text{ in } \Omega}} \left| \frac{\partial^2 g(y)}{\partial y_i^2} \right|.$$

The bound is sharp for the function $g(z) = z^T(e - z)$; in this case $G_M(z) \equiv 0$. Using a simple scaling argument, one obtains

THEOREM 2.2. *For every $z = (x_1, \ldots, x_N)^T$ within the table limits,*

$$|(f - F_M)(z)| \le \frac{N}{8} h^2 \sup_{\substack{i=1,\ldots,N \\ x}} \left| \frac{\partial^2 f(x)}{\partial x_i^2} \right|,$$

where $h = \max_{i,j} |X_{i_j+1} - X_{i_j}|$.

3. A Piecewise Linear Interpolant. The piecewise linear interpolant F_L can be computed by breaking the unit N-cube into the $N!$ simplices [1], [7] of the form

$$S_{p(1),\ldots,p(N)} = \{(y_1, \ldots, y_N) : 0 \le y_{p(1)} \le \cdots \le y_{p(N)} \le 1\},$$

where $(p(1), p(2), \ldots, p(N))$ is a permutation of the integers $1, \ldots, N$. The particular simplex of interest is found by sorting the coefficients $y_1, \ldots, y_N$ of the evaluation point. F_L is of the form

$$a_0 + a_1 y_1 + \cdots + a_N y_N$$

in each simplex. The coefficients $\{a_i\}$ are defined so F_L matches f at the $N + 1$ corners $s_0, \ldots, s_N$ of the simplex, where s_i has 1 in positions $p(j)$, $j > i$, and 0 elsewhere. F_L can be computed as follows:

sort $\{y_i\}$ to determine $\{p(i)\}$
$s_0 = (1, \ldots, 1)^T$
$F_L := g(s_0)$
for $i = 1$ to N
  $s_i = s_{i-1} - e_{p(i)}$
  $F_L := F_L + (1 - y_{p(i)})(g(s_i) - g(s_{i-1}))$
next i,

I think this is what Ryan does in ndim_simplex.m

where $e_{p(i)}$ has a 1 in position $p(i)$ and 0 everywhere else. For example, Figure 2 indicates the case $N = 3$, $0 \le y \le z \le x \le 1$.

The gradient ∇F_L can also be evaluated cheaply:

for $i = 1$ to N

$$F_L^{x_{p(i)}} = \frac{g(s_{i-1}) - g(s_i)}{X_{I_{p(i)}+1} - X_{I_{p(i)}}}$$

next i.

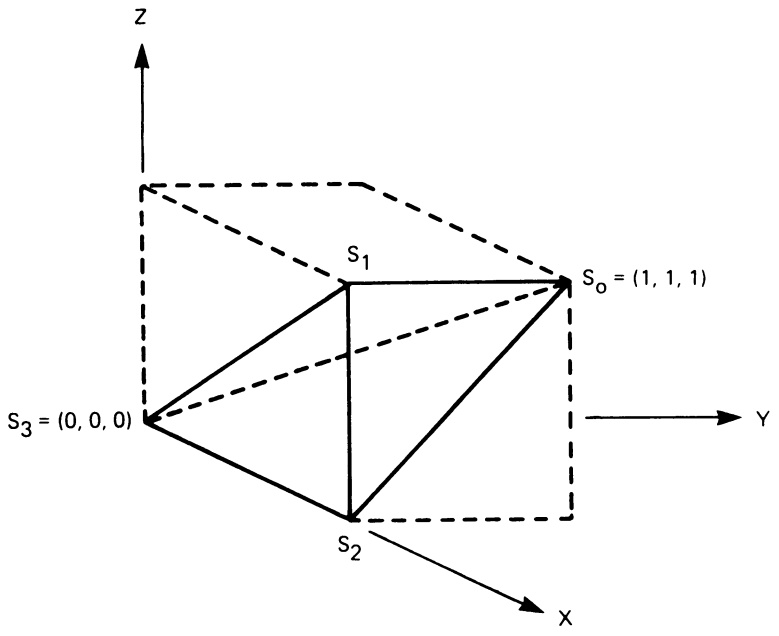

FIGURE 2. S_{231}

A pointwise error bound for $F_L = G_L$ is as follows:

THEOREM 3.1. *For every $z = (y_1, \ldots, y_N)^T$ in the unit N-cube Ω,*

$$|(g - G_L)(z)| \leq z^T(e - z)|||g|||/2,$$

where $e = (1, \ldots, 1)^T$ and the seminorm $|||g|||$ is defined by

$$|||g||| = \sup_{\substack{\theta^T\theta=1 \\ y \text{ in } \Omega}} |D_\theta^2 g(y)|,$$

where $D_\theta g(y)$ is the directional derivative

$$D_\theta g(y) = \lim_{h \to 0} \frac{g(y + h\theta) - g(y)}{h}.$$

Proof. In barycentric coordinates,

$$z = \sum_{i=0}^{N} c_i s_i,$$

where $0 \leq c_i \leq 1$ for all i and $\sum_{i=0}^{N} c_i = 1$. By Taylor's theorem, for each i,

$$g(s_i) = g(z) + (\nabla g(z))^T(s_i - z) + (s_i - z)^T(s_i - z)D_\theta^2 g(\xi_i)/2$$

for some ξ_i on the line segment between s_i and z, and

$$\theta = (s_i - z)/((s_i - z)^T(s_i - z))^{1/2}.$$

Then

$$G_L(z) = \sum_{i=0}^{N} c_i g(s_i) = \sum_{i=0}^{N} c_i g(z) + \sum_{i=0}^{N} (\nabla g(z))^T c_i (s_i - z)$$
$$+ \sum_{i=0}^{N} c_i D_\theta^2 g(\xi_i)(s_i - z)^T (s_i - z)/2$$
$$= g(z) + \sum_{i=0}^{N} c_i D_\theta^2 g(\xi_i)(s_i - z)^T (s_i - z)/2.$$

Therefore,

$$|(g - G_L)(z)| \le \sum_{i=0}^{N} c_i (s_i - z)^T (s_i - z) |||g|||/2.$$

By symmetry, it now suffices to consider the simplex

$$S_{1,\dots,N} = \{(y_1, \dots, y_N) : 0 < y_1 < \cdots < y_N = 1\}.$$

It can be verified by induction that in this simplex,

$$c_0 = y_N, \quad c_i = y_{N-i} - y_{N-i+1}, \quad c_N = 1 - y_1$$

and

$$(s_0 - z)^T (s_0 - z) = (1 - y_1)^2 + \cdots + (1 - y_N)^2,$$
$$(s_i - z)^T (s_i - z) = (1 - y_1)^2 + \cdots + (1 - y_{N-i})^2 + y_{N-i+1}^2 + \cdots + y_N^2,$$
$$(s_N - z)^T (s_N - z) = y_1^2 + \cdots + y_N^2,$$

for $0 < i < N$. Cancelling terms,

$$\sum_{i=0}^{N} c_i (s_i - z)^T (s_i - z) = z^T (e - z). \quad \square$$

The bound is sharp for the function $g(z) = z^T(e - z)$; in this case $G_L(z) \equiv 0$. This theorem could also have been proved using direct induction in the original coordinates: The resulting θ's would only occur in directions parallel to simplex edges.

As before, by a scaling argument, one has

THEOREM 3.2. *For every $z = (x_1, \dots, x_N)^T$ within the table limits,*

$$|(f - F_L)(z)| \le \frac{N}{8} h^2 \sup_{\substack{\theta^T \theta = 1 \\ x}} |D_\theta^2 f(x)|.$$

4. Discussion. We tested the relative computer times to evaluate F_M and F_L on an IBM 3081 computer using single precision and a simple bubble sort for F_L. In this environment, floating-point multiplies and compares took about the same time. We found that once the intervals of interest had been selected, evaluating F_M took about twice the time as F_L for $N = 4$, and about 27 times the time for $N = 10$.

Except for the norms on g, the pointwise error bounds for the two interpolants are identical, and the two interpolants are usually of comparable accuracy. Both interpolants are continuous and monotone. F_L is continuous because the points

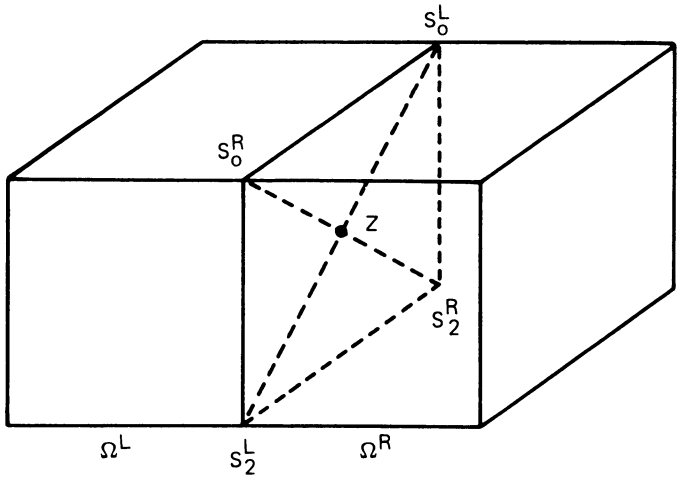

FIGURE 3. $F_L(z^L) \neq F_L(z^R)$

$\{s_0\}$ are chosen in a consistent way. Global continuity of F_L would not hold if the point s_0 in each N-cube was chosen independent of orientation (Figure 3).

Each evaluation of F_L depends on $N+1$ table entries, while each evaluation of F_M depends on 2^N table entries. Thus, using F_L can require many fewer function evaluations if table entries are only computed when needed. Since ∇F_L is constant in each simplex, Newton's method applied to a vector function with components of the form of F_L converges in one step after the simplex containing the solution is reached (cf. [1]).

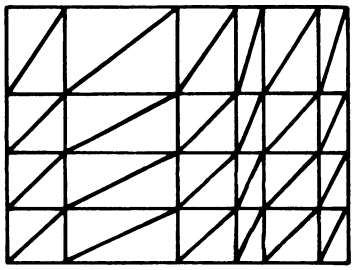

FIGURE 4a

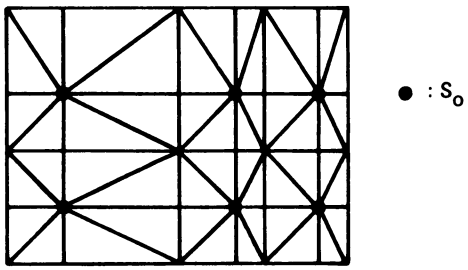

FIGURE 4b

F_L has a systematic preference in each N-cube for the direction from s_N to s_0 (Figure 4a). For instance, the value at the center of the N-cube depends only on the table values at s_0 and s_N. A variant that reduces the directional preference and maintains global continuity chooses $s_0 = (i(1), \ldots, i(N))$, where each $i(j)$ is chosen so that the global index $I_j + i(j)$ is even (Figure 4b). The resulting interpolant, in effect, changes coordinates from y_j to $1 - y_j$ whenever $i(j) = 0$.

Exxon Production Research Company
P.O. Box 2189
Houston, Texas 77252-2189

1. E. ALLGOWER & K. GEORG, "Simplicial and continuation methods for approximating fixed points and solutions to systems of equations," *SIAM Rev.*, v. 22, 1980, pp. 28–85.

2. C. DE BOOR, *A Practical Guide to Splines*, Springer-Verlag, New York, 1978, pp. 91–93.

3. G. DAHLQUIST & A. BJÖRCK, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, N. J., 1974, pp. 319, 323.

4. W. A. DAHMEN & C. A. MICCHELLI, "On the linear independence of multivariate B-splines, I. Triangulations of simploids," *SIAM J. Numer. Anal.*, v. 19, 1982, pp. 993–1012.

5. P. F. DUBOIS, "Swimming upstream: Calculating table lookups and piecewise functions," in *Parallel Computations* (G. Rodrigue, ed.), Academic Press, New York, 1982, pp. 129–151.

6. K. HÖLLIG, "Multivariate splines," *SIAM J. Numer. Anal.*, v. 19, 1982, pp. 1013–1031.

7. H. W. KUHN, "Some combinatorial lemmas in topology," *IBM J. Res. Develop.*, v. 45, 1960, pp. 518–524.

8. M. H. SCHULTZ, *Spline Analysis*, Prentice-Hall, Englewood Cliffs, N. J. 1973, pp. 10–20.