Algorithm 791: TSHEP2D: Cosine Series Shepard Method for Bivariate Interpolation of Scattered Data

ROBERT J. RENKA
University of North Texas
and
RON BROWN
AISN Software

We describe a new algorithm for scattered data interpolation. It is based on a modified Shepard method similar to that of Algorithm 660 but uses 10-parameter cosine series nodal functions in place of quadratic polynomials. Also, the interpolant has continuous second partial derivatives. An accompanying survey article presents test results that show the method to be more accurate than polynomial-based methods in terms of reproducing test functions with large variations and steep gradients.

Categories and Subject Descriptors: D.3.2 [**Programming Languages**]: Language Classifications—*Fortran 77*; G.1.1 [**Numerical Analysis**]: Interpolation; G.1.2 [**Numerical Analysis**]: Approximation; G.4 [**Mathematics of Computing**]: Mathematical Software

General Terms: Algorithms

Additional Key Words and Phrases: Interpolation, scattered data, Shepard method, surface fitting

1. METHOD

We treat the problem of constructing a smooth bivariate function C that interpolates data values f_k at scattered nodes (x_k, y_k) in the plane for $k = 1, \ldots N$. We employ a modified Shepard method with a cell-based search algorithm as described in Renka [1988a]. The interpolant is defined by

$$C(x, y) = \sum_{k=1}^{N} W_k(x, y) C_k(x, y) \Bigg/ \sum_{i=1}^{N} W_i(x, y),$$

Authors' addresses: R. J. Renka, Department of Computer Sciences, University of North Texas, Denton, TX 76203-1366; email: renka@cs.unt.edu; R. Brown, AISN Software, P.O. Box 449, Mapleton, OR 97453.

where the nodal function C_k is a bivariate cosine series that interpolates the data value f_k at node k and fits the data values on a set of nearby nodes in a weighted least-squares sense.

The unnormalized weights are inverse distance functions:

$$W_k(x, y) = \left[\frac{(R_w - d_k)_+}{R_w d_k}\right]^3$$

for

$$(R_w - d_k)_+ = \begin{cases} R_w - d_k & \text{if } d_k < R_w \\ 0 & \text{if } d_k \geq R_w \end{cases},$$

where $d_k(x, y)$ is the Euclidean distance between (x, y) and (x_k, y_k), and R_w is a radius of influence about (x_k, y_k). An interpolated value at a point (x, y) depends only on the data at nodes whose radii include (x, y).

It follows from the above definition that C interpolates the data, maintains the local shape properties of the nodal functions (has first and second partial derivatives at (x_k, y_k) that agree with those of C_k), and lies in the space $C^2(\mathbf{R}^2)$.

Nodal function C_k is defined by

$$\begin{aligned} C_k(x, y) = {} & a_{1k} + a_{2k}\cos(p) + a_{3k}\cos(q) + a_{4k}\cos(2p) \\ & + a_{5k}\cos(p)\cos(q) + a_{6k}\cos(2q) + a_{7k}\cos(3p) \\ & + a_{8k}\cos(2p)\cos(q) + a_{9k}\cos(p)\cos(2q) + a_{10k}\cos(3q), \end{aligned} \tag{1}$$

for

$$p = \frac{x - \text{Xmin}}{\text{Xmax} - \text{Xmin}}\pi$$

and

$$q = \frac{y - \text{Ymin}}{\text{Ymax} - \text{Ymin}}\pi,$$

where Xmin, Xmax, Ymin, and Ymax are the extrema of the nodal coordinates, so that $(x, y) \in$ [Xmin,Xmax] $\times$ [Ymin,Ymax] implies $(p, q) \in [0, \pi] \times [0, \pi]$. The coefficients are obtained by a weighted least squares fit to the data values at node k and the closest nodes to k: the solution to minimizing

$$\sum_{i=1}^{N} \omega_{ik}[C_k(x_i, y_i) - f_i]^2$$

for

$$\omega_{ik} = \left[\frac{(R_c - d_{ik})_+}{R_c d_{ik}} \right]^2,$$

where d_{ik} is the distance between nodes i and k, and R_c is a radius of influence about node k; C_k depends only on the data values at nodes within distance R_c of (x_k, y_k). The weight ω_{kk} associated with node k is taken to be a large value so that C_k interpolates f_k at node k.

The radii R_c and R_w vary with k and are taken to be just large enough to include N_c and N_w nodes, respectively, for fixed values of N_c and N_w. The optimal values of these parameters depend on the data set, but accuracy varies smoothly and gradually with variations in the values. The default recommendations, found to be optimal for a set of test cases, are $N_c = 18$ and $N_w = 32$. Slightly smaller values may produce better results on sparse data sets.

Note that, in general, the support of C is the union of a set of node-centered disks with radii that depend on N_w. If the nodal density varies widely, this union of disks may not cover the convex hull of the nodes, i.e., the convex hull could include points (x, y) for which $C(x, y) = 0$ because (x, y) is not within the radius of influence of any node. Thus, it may be necessary to use a larger value of N_w in order to avoid this situation.

The cell-based search method is used in the preprocessing phase to determine an ordered sequence of nearest neighbors to each node, and in the evaluation phase to find the set of all nodes whose radii R_w include the evaluation point. The smallest rectangle containing the nodes is partitioned into an $N_r \times N_r$ uniform grid of cells, and the indexes of the nodes contained in each cell are stored as a linked list in two integer arrays. For maximum efficiency, the recommended value of N_r is $\sqrt{N/3}$. Assuming a uniform distribution of nodes, the expected time complexity is $O(N)$ for the preprocessing phase and constant for each evaluation. Worst-case operation counts are $O(N^2)$ for preprocessing and $O(N)$ for evaluation.

The accompanying survey article [Renka and Brown 1999] presents test results showing that, for difficult test funtions, TSHEP2D is among the most accurate scattered data algorithms available.

2. CODE

The software is written in 1977 ANSI Standard Fortran and uses double precision. It can be converted to single precision by simply replacing all occurrences of 'DOUBLE PRECISION' or 'DBLE' by 'REAL'. Note, however, that there is a significant amount of roundoff error, particularly in evaluating partial derivatives, and IEEE standard single precision is not sufficient. There are no system dependencies. The array storage requirements consist of three order-N arrays, X, Y, F, containing the data points, a $10 \times N$ array

A for the coefficients, an array RW of length N for the weights W_k, an $N_r \times N_r$ integer array LCELL for the index of the first node in each cell, and an integer array LNEXT of length N for next-node indexes.

The code is modularized in a fashion similar to that of Algorithm 660 [Renka 1988b]. The user-callable subprograms are as follows:

TSHEP2 Subroutine which computes the parameters defining the interpolant C.

TS2VAL Function which returns the value of C at an arbitrary point.

TS2GRD Subroutine which returns the value and gradient of C at an arbitrary point.

TS2HES Subroutine which returns the value, gradient, and Hessian of C at an arbitrary point.

STORE2 Subroutine which computes and stores the data structure for cell-based searches.

GETNP2 Subroutine which returns the nearest unmarked node, along with its Euclidean distance, to an arbitrary point, and marks the node (so that a subsequent call will return the next closest node).

TSHEP2 calls STORE2 and GETNP2 to find sequences of nearest neighbors to each node. It calls three additional subroutines to set up and solve the least-squares systems for the coefficients defining C. There are no other subprogram dependencies. Subroutines STORE2 and GETNP2 could be extracted from the source code and used to solve more general closest-point problems.

REFERENCES

Renka, R. J. 1988a. Multivariate interpolation of large sets of scattered data. *ACM Trans. Math. Softw. 14*, 2 (June 1988), 139–148.

Renka, R. J. 1988b. Algorithm 660: QSHEP2D: Quadratic Shepard method for bivariate interpolation of scattered data. *ACM Trans. Math. Softw. 14*, 2 (June 1988), 149–150.

Renka, R. J. and Brown, R. 1999. Algorithm 792: Accuracy tests of ACM algorithms for interpolation of scattered data in the plane. *ACM Trans. Math. Softw. 25*, 1 (Mar.). This issue.

Received: March 1997; accepted: July 1998