W. R. Tobler

Lattice Tuning[1]

A method of spatial interpolation from an arbitrary arrangement of numerical observations is derived, with examples shown as contour maps. The fit to the original data and the smoothness of the resulting lattice values are optimized simultaneously by an iterative numerical technique. The algorithm does not require specification of distance dependent influence functions.

Given a scatter of numerical values at irregularly arranged locations on a plane (X_k, Y_k, Z_k, k=1,…,n)*,* it is common practice to interpolate to values at a regular lattice. These lattice estimates are then used in subsequent processing to, for example, draw contour maps. There are many ways of performing the interpolation to the lattice points {6, 11, 18, 19, 22, 28, 31]. In many of these methods a form of inverse distance weighting is used to assign estimated values to the mesh intersections from the nearby observational data points. Alternate definitions of nearby, and variations in the weighting functions, result in different estimates at the lattice points. One method of comparing these alternate algorithms is to interpolate back to the original observations by using the estimated lattice point values. In general some discrepancy is found if one performs this reinterpolation. A natural measure by which to compare alternate interpolations is then by the use of the root-mean-squared error of such a backward interpolation. But it is fairly obvious that this method can also be used to improve the estimates at the lattice points, i.e., to minimize the discrepancy of the reverse interpolation. If the back-estimate is too low (high), for example, it is only necessary to raise (lower) the values at the controlling lattice points. I refer to this as a tuning of the lattice. The method is iterative and works by adding adjustments to lattice points that surround the observations. The method can also be used to do the complete interpolation by assigning arbitrary initial estimates to the lattice and by then systematically improving these estimates. As an interpolation method per se, the tuning has the additional advantage that one does not have to perform the expensive inter-point distance computation that is required for many of the weighted-neighbor interpolation methods.

One, instead, sorts the observations into the cells of the lattice, an n ln process.

NOTATION AND AN EXAMPLE

It is simplest to begin by assuming that the lattice geometry has already been established and cannot be changed, and that linear interpolation is used within the lattice. Denote the four vertices of the lattice by the subscripts (0,0), (0,1), (1,0), (1, 1), and define a right-handed system of Cartesian coordinates for the location of the observations. The lattice cell is assumed to be rectangular and an observation with value Z is positioned at the location x, y within this cell. The interpolated value, denoted by Z*, is obtained by estimation from the four corners of the lattice. The bilinear interpolation equation is

$$Z^* = \frac{1}{\Delta x \Delta y} \sum_{p=0}^{1} \sum_{q=0}^{1} W_{pq} Z_{pq},$$

where W_{pq} = P(x) Q(y), with

$$\begin{aligned} P(x) &= \Delta x - (x - x_0) && \text{when } p = 0; \\ &= x - x_0 && \text{when } p = 1; \\ Q(y) &= \Delta y - (y - y_0) && \text{when } q = 0; \\ &= y - y_0 && \text{when } q = 1; \end{aligned}$$

and Z_{pq} is the value at the (p, q) lattice point. The lower left corner of the lattice cell has coordinates x_o, y_o. The cell is Δx by Δy in size; $\Delta x = x_1 - x_0$, $\Delta Y = y_i - y_o$. This notation may seem excessive for the simple bilinear weighting required for interpolation within a square, but it has later advantages, and it generalizes in a simple manner to higher dimensions.

As a numerical example, suppose that the values at the four lattice points are as follows:

$$Z_{oo}=12, Z_{01}=10, Z_{10}=15, Z_{11}=20,$$

and that $x_o = 1$, $y_o = 1$, $\Delta x = 4$, $\Delta y = 4$, and that there is an observation $Z = 15.0$ at the location $x = 3$, $y = 2$. Then one finds that $Z^* = 13.875$ and the discrepancy is $15.0 - 13.875 = 1.125$. This difference can be eliminated by changing the values at the four corners, in this case pulling them up, since the estimate is too low. To obtain an exact fit it is necessary to change each corner value by the amount δ_{pq} to satisfy

$$Z = Z^* + \Delta$$

$$= \frac{1}{\Delta x \Delta y} \sum_{p=0}^{1} \sum_{q=0}^{1} W_{pq}(Z_{pq} + \delta_{pq}).$$

The simplest solution is to take $\delta_{pq} \equiv \Delta$, an equal increment for all four corners. The new estimate would then in fact yield the required value. But this is not entirely satisfactory. One would expect that lattice points close to the observation should receive a greater correction than distant ones. Also, if there is more than one observation in a cell, and one were overestimated, the other underestimated, a simultaneous adjustment could never be effected. The increment to the corners should take into account the position of the observation within the cell. The corner-point values will, of course, also be affected by observations in adjacent rectangles.

The one constraint that must be satisfied is

$$\Delta = \frac{1}{\Delta x \Delta y} \sum_{p=0}^{1} \sum_{q=0}^{1} W_{pq} \delta_{pq},$$

but there are four unknowns. Aside from an intuition that the corrections should be similar to the weights, the appropriate choices for δ_{pq} are not obvious, and there are, in fact, several possibilities. If one assumes that the initial lattice values are reasonable, then one might wish to modify these values by as little as possible. Thus, writing the foregoing equation in vector notation as

$$\Delta = (1/k) W^t \delta,$$

and the least-squares solution is then obtained using the pseudo-inverse [20, p. 39) as

$$\delta = k\Delta W/(W^t W),$$

where k is the product ΔxΔy. Using the previous example, new values for the lattice points are obtained:

$$Z_{oo} = 13.35, Z_{10} = 16.35, Z_{01} = 10.45, Z_{11} = 20.45,$$

and the linear interpolation within the lattice now gives Z* = Z = 15, as desired.

SMOOTH INTERPOLATION

The foregoing solution is very dependent on a satisfactory set of initial estimates, and the adjustments are chosen so as to avoid radical departures from these estimates. But this does not offer any suggestions for interpolation *ab initio*. The initial assignment might consist of zeros, of the mean value of the scalar, of the value at the nearest observation, of a trend surface fit, of a weighted moving average, or of a covariance or varigram estimate, etc. Some of these initial estimates for the lattice values are clearly better than others. Thus, the requirement that the adjustment is to yield values that are near to the previous values can hardly be defended.

Instead of placing such heavy reliance on these initial estimates, let **it** be required that the interpolation result in values at lattice points that match the observations, but that are also "smooth." This has particular merit if one later wishes to use derivatives of the lattice estimates. A completely smooth surface would be obtained if the value at each lattice point were equal to the average of the values from the neighboring lattice points. The adjustments can be chosen to minimize the departure from this target value. It is arguable that some other side condition might be better, for example matching some covariance or varigram structure, but this possibility is not explored here; for developments along these lines see [7, 13, 17,23,24]. The smooth interpolation proceeds by making estimates at each lattice point, computed as a weighted average of the values at the neighboring lattice points, and the least-squares adjustment is then applied using these estimates as the target. This, incidentally, also demonstrates that smoothing adjustments can take place even when there are no discrepancies between values at the observational locations and their interpolations from the lattice. Each application of the least-squares adjustment changes all of the lattice values surrounding an observation, and new averages can now be computed for the neighbors. The entire operation is then repeated in an iterative fashion. The direct averaging of neighboring lattice values can be shown to be the equivalent of a finite representation using Laplace's equation. Averaging of the derivatives approximated on the lattice yields even smoother transitions and leads to a representation using the biharmonic equation [15]. Further justification for the use of the biharmonic equation is given in [3]. Iterative solutions of both the Laplace equation and the biharmonic equation are known to be uniquely convergent when the boundary conditions are specified [5, vol. 2, chap. 4].

The discussion has thus far focused on a single lattice cell. But a lattice point, unless it is at the edge of the region, will be associated with four cells, each of which may demand a change in the value at that lattice point. Changing the value of one corner of a cell will also propagate effects to the other corners of that cell. These consequences are covered by using an iterative procedure. That is, small changes are made to lattice points on one computer pass through the entire array. These estimates are then improved on subsequent passes. The neighbor-averaging and least-squares tuning process is a stable and converging one that improves the estimates and the smoothness on every pass. Convergence can, in fact, be hastened by taking overlapping neighborhoods in setting the desired lattice target values.

Some lattice points will be associated with mesh cells that do not contain any observations. Estimates can be made at these locations, since the connectivity of all of the lattice points to each other is known. It is this lack of knowledge of adjacencies that makes interpolation from randomly scattered observations difficult. Lattice points that are not affected by the tuning are relaxed with the tuned values as constraints. The process may most easily be visualized as follows. After an initial assignment has been made for each mesh point, replace this number by the value that it would receive by interpolation from its immediate neighbors. Do this in some regular sequence so that, after an interpolation is made at one point, this interpolated value is used in the interpolation of the value at the neighboring point. In this manner every point affects its neighbors and is affected by them. This bootstrap operation pulls along all of the values to which it is applied. After passing over the array in this manner several times a stable situation will have been reached in which the interpolated value at a point is sensibly the same as the value obtained on the previous pass. Neighboring points will then have values that are similar, and it is time to stop. The actual observations in this process are fixed points that affect their neighbors, but that are not affected by them. Fortunately the initial guess is not critical; the target function is. Either linear or curvilinear interpolation can be used to tune the lattice, but any method of interpolation must be recognized as a hypothesis about nature, which may or may not be true. The present implementation uses linear interpolation.

The method described requires that an estimate be made at each lattice point by averaging from the neighboring lattice points. Edge points have only a restricted number of lattice neighbors and a different strategy must be adopted in their vicinity. In fact, the smoothness criterion chosen yields a finite difference approximation to either Laplace's equation or the biharmonic equation where the observations play the role of "sources." These two equations are known in the technical literature as elliptical partial differential equations, and the iterative procedure reported here coincides with methods for solving such equations; see [2, 9, 15, 32]. These works also treat the boundary problem *in extensio*. Briefly, one must use some *a priori* method of either assigning a value to lattice points along the edge of the domain, or one must assign a value to the normal derivative at these edges. I have used the latter procedure by fitting a least-squares plane to the entire set of data, and then setting the derivative at the edges of the domain equal to the slopes of this plane. The other obvious alternative is to use an inverse distance estimate involving points near the edge. This is the strategy derived by Brigg [3] and programmed by Swain [30] using equations very similar to those used here. A theoretically elegant, but computationally tedious, method of assigning values along the edges is described in [14, pp. 163-64].

The foregoing steps can all be extended to three-dimensional interpolation, to the interpolation of vector or tensor field components, to observations on a sphere, or to alternate lattice geometries. Occasionally, for example, a triangular lattice, rather than a rectangular one, is used. In this case the optimal criterion would seem to be to choose the adjustments at the three vertices so that the normal to the surface patch is, as nearly as possible, parallel to a weighted linear combination of the surface normals of the three neighboring triangular facets, while retaining the objective that reinterpolation within each triangle yields the observed value. The triangular lattice is not to be confused with a triangulation in which observations are located at the vertices of the triangles, as for example in [1, 10].

THE LATTICE SIZE

A disadvantage of the proposed method for interpolation is that **it** is necessary to choose a lattice size. The literature reviewed is embarrassingly quiescent on this point. But the sampling theorem [26] suggests that the spacing must be smaller than one-half the size of the smallest feature to be detected. An estimate of this smallest feature is obtainable by considering the highest degree of trigonometric polynomial that could be calibrated from the data. This degree cannot exceed n, the number of observations. In k dimensions the effective number of observations is the kth root of n. Thus an upper bound on the size of the lattice interval should be the kth root of the volume of space occupied by each observation. This corresponds to the resolution of the data. In two dimensions it becomes $\Delta y = \Delta x \leq (\text{area}/n)^{½}$, which has units of length per effective observation. The values used in the examples to be shown conform to this. The number of lattice points is then also approximately equal to the number of actual observations. It is not efficient to stray too far from this value, but the spatial arrangement of the observations may exhibit clustering so that the exact limit is too lax. A lattice spacing that is two-thirds of the foregoing limit has been found effective when the observations lie within a square region.

In the current computer implementation of tuning, a perfect fit to the data cannot be obtained when there are several observations inside of one mesh cell. As the mesh becomes smaller these discrepancies of course disappear, but the computational cost increases. It is always possible to fit the data to an orthogonal, irregularly spaced n by n mesh with observations at n nodes, leaving n^2 - n nodes to be assigned by relaxation. A working computer program demonstrates this to be very effective, but it is practical only for small n, small being defined by the size of one's computer. A description of this alternate method is being prepared for publication.

Examples

My experimentation has included variation in the initial estimates, changes in the boundary conditions and in the target function, and in application to a variety of actual data, usually of small size - less than 500 observations. In one experiment the initial values at lattice points consisted of (1) the mean of all observations, (2) random numbers, (3) marginal sums, or (4) values assigned by a conventional inverse distance weighting. The four configurations were virtually identical after only a few iterations in spite of these disparate initial estimates. A more realistic practical evaluation is provided by the 100-observation example discussed in [4] and in [21] where the data are given. Twenty iterations of the algorithm on an 11 by 11 lattice yielded the map shown in Figure 1. In this instance, reinterpolation from the lattice to the original observations can be interpreted as a model that accounts for 95 percent of the original variance. A slight improvement is obtained using the biharmonic equation as the target. I also present an example using data listed by Akima [1]. The contour map (Fig. 2) is drawn using linear interpolation within the 143 (= 11 x 13) point lattice tuned from the fifty original observations. The mean of the row and column marginals were used as the initial estimates and twenty iterations were performed with the biharmonic as the target. The model accounts for approximately 99.9 percent of the variance. This small lattice could be expanded to a larger lattice by splining for cosmetic effects, which of course has nothing to do with the tuning process. But the tuning can also be considered an implicit form of splining directly from irregularly arranged observations; the present work may thus be compared with [8, 16, 20]. A final map shows a result using a smaller grid interval. Here the lattice is 39 by 49 and only one mesh point is directly influenced by more than one observation. In other words no cell contains more than one observation and 97 percent contain no observations. Most of the computational effort in this excessively fine grid is devoted to the filling of vacancies between the observations. The resulting contours are quite smooth, and all of the variance (to two decimal points) is explained by the values assigned to the mesh; the fit to the data is essentially perfect.

LITERATURE CITED

1. Akima, H. “A Method of Bivariate Interpolation and Smooth Surface Fitting for Values Given at Irregularly Distributed Points.” Washington, D.C. Office of Telecommunications Report, 70-75, 1975.
2. Birkhoff, C. *The Numerical Solution of Elliptic Equations.* Philadelphia: SIAM, 1972.
3. Briggs, L “Machine Contouring Using Minimum Curvature.” *Geophysics,* 89 (1974), 39-48.
4. Cole, A. “Algorithm for the Production of Contour Maps from Scattered Data.” *Nature,* 220 (1968), 92-94.
5. Courant, B. and D. Hilbert. *Methoden der Mathematischen Physik.* Berlin: J. Springer, 1937.
6. Cram, J. “Computer Interpolation and Contouring of Two Dimensional Data: A Review.” *Geoexploration,* 8 (1970) 71-86.
7. David, M. *Geostatistical Ore Reserve Estimation.* New York: Elsevier, 1977.
8. Dooley, J. “Two Dimensional Interpolation of Irregularly Spaced Data Using Polynomial Splines.” *Physics of the Earth and Planetary Interiors,* 12 (1978), 180-187.
9. Forsythe C., and W. Wasow. *Finite Difference Methods for Partial Differential Equations.* New York: Wiley, 1980.
10. Gold, C., T. Charters, and J. Ramsden. “Automated Contour Mapping Using Triangular Element Data.” *Computer Graphics,* 11(1977), 170-175.
11. Grafarend, E. “Geodetic Prediction Concepts.” *Methoden und Verfahren der mathematischen Physik,* 13 (1975), 161-200, (Mathematical Geodesy, part 2.) Zurich: Bibliographisches Institut.
12. Greene, G., S. Outcalt, and W. Tobler. “Reconstruction and Parsing of Topographic Data.” In preparation.
13. Harbaugh, J., J. Doreton, and J. Davis *Probability Methods in Oil Exploration.* New York: Wiley, 1976.

14. Hockney, B. W. “The Potential Calculation and Some Application.” In *Methods in Computa-tional Physics,* vol. 9, edited by Alder, Fernback, and Rotenberg, pp. 135-211. New York Academic Press, 1970.

15. Kantorovich, L., and V. Krylov. *Approximate Methods of Higher Analysis.* The Hague: Noordhoff Interscience, 1958.

16. Knudson, W., and D. Nagy. “Discrete Data Smoothing by Spline Interpolation with Application to Initial Geometry of Cable Nets.” *Computer Methods in Applied Mechanics and Engineering,* 4 (1974), 321-48.

17. Kriege, D., “A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand.” *J. Chem. Metall. Min. Soc. S. Afr.,* 52 (1951), 119-39.

18. Lancaster, P., and K. Salkaulkas. “A survey of Curve and Surface Fitting.” Unpublished manuscript, University of Calgary, 1977.

19. Lawson, C. “Software for C’ Surface Interpolation.” In *Mathematical Software III,* New York: Academic Press, 1978.

20. Lawson, C., and R. Hanson. *Solving Least Squares problems.* Englewood Cliffs: Prentice-Hall, 1974.

21. Mcintyre, D., et al. *Computer Programs for Automatic Contouring.* Comp. Contrib. No. 23, Lawrence, Kansas Geological Survey, 1968.

22. Mansfield, L. “Optimal Approximation and Error Bounds in Spaces of Bivariate Functions.” *J. Approx. Theory,* 5(1972), 77-96.

23. Matheron, G., *The Theory of Regionalized Variables and its Applications.* Cahiers du Centre de Morphologie Mathematique de Fontainbleau: Ecole National Superieure des Mines de Paris, 1971.

24. Moritz, H., *Eine Allgemeine Theorie der Verarbeitung von Schwermessungen nach Kleinsten Quadraten.* Heft Nr. 67k Deutsche Geodatische Komission, Munich, 1970.

25. Moritz, “Elements of Collocation.” *Methoden und Verfahren der mathematischen Physik,* 13 (1975), 125-59. (Mathematical Geodesy, part II;) Zurich: Bibliographisches Institut.

26. Petersen, D., and D. Middleton. “Sampling and Reconstruction of Wavenumber Limited Functions in N-Dimensional Euclidean Spaces.” *Information and Control,* 5(1962), 279-

323.

27. Schumaker, L. “Fitting Surfaces to Scattered Data.” In *Approximation Theory II,* edited by C. Lorentz et al., pp. 203-68. New York: Academic Press, 1976.

28. Schut, C. “Review of Interpolation Methods for Digital Terrain Models.” *The Canadian Surveyor,* 30 (1976), 389-412.

29. Southwell, R. *Relaxation Methods in Theoretical Physics.* Oxford: Clareadon Press, 1956.

30. Swain, C. “A Fortran 1V Program for Interpolating Irregularly Spaced Data Using the Difference Equations for Minimum Curvature.” *Computers and Geosciences,* 1 (1976), 231-240.

31. Tobler, W. “Linear Operators Applied to Areal Data” In *Display and Analysis of Spatial Data,* edited by J. Davis and M. MacCullogh, pp. 14-37. New York: Wiley, 1975.

32. Wachspress E. *Iterative Solution of Elliptic Systems.* Englewood Cliffs: Prentice-Hall, 1966.

[1]**GEOGRAPHICAL ANALYSIS,** vol. 11, no. 1 (January 1979)

0016-7363/79/0179-0036$00.50/0 © 1979 Ohio State University Press

W. R. Tobler is professor of geography, University of California, Santa Barbara.

Submitted 5/78. Accepted 6/78.

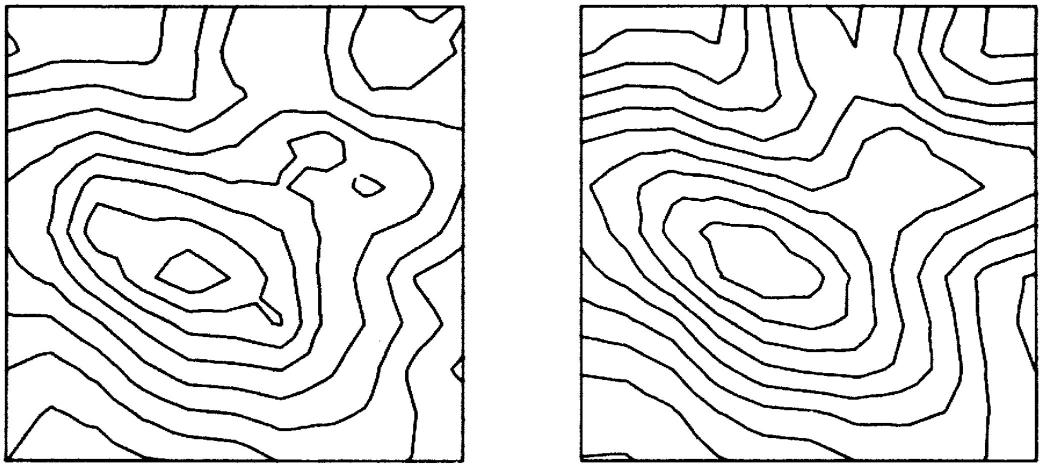

FIG. 1. Interpolation, by Tuning, of an 11 by 11 Lattice Superimposed on 100 Randomly Arranged Scalars, after Twenty Iterations from the Average of the Row and Column Marginals, Then Contoured by Linear Interpolation. Left: adjusted to Laplace's equation; right: adjusted to the biharmonic equation. Compare with [*4*, *21*], where the same data are used. The cost of the tuning was $1.75 on the University of Michigan Amdahl 470V/6 computer.

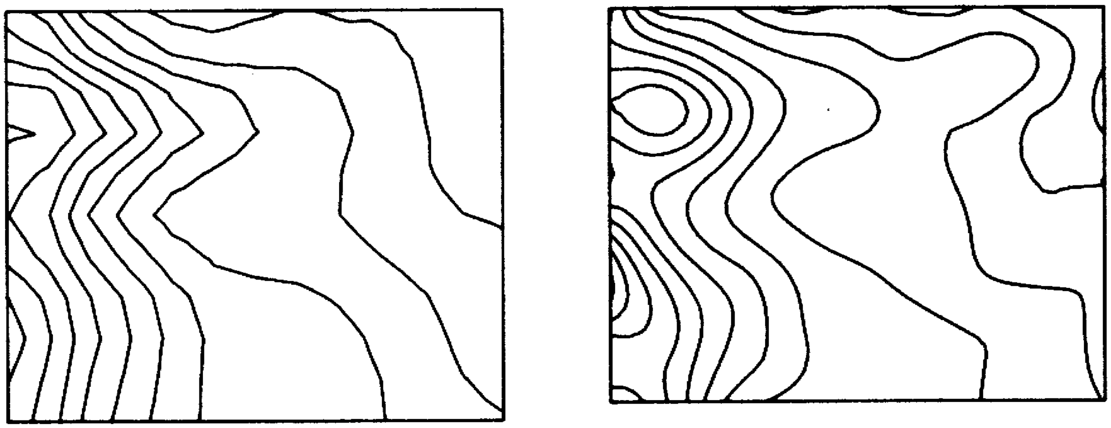

FIG. 2. Contouring, by Linear Interpolation, of 50 Randomly Arranged Observations Listed in [*1*]. Left: an 11 by 13 lattice tuned using 20 iterations starting from the average of the row and column marginals with the biharmonic as target. Right: the same but using a 39 by 49 lattice and 55 iterations.