

# PACIFIC SOUTHWEST Forest and Range Experiment Station

FOREST SERVICE  
U.S. DEPARTMENT OF AGRICULTURE  
P.O. BOX 245, BERKELEY, CALIFORNIA 94701

## Interpolation of unevenly spaced data using a parabolic leapfrog correction method and cubic splines

Julio L. Guardado

William T. Sommers

*USDA Forest Service  
Research Note PSW-324  
1977*

Data collection sites used during field experiments or in operational observing networks are usually not located at points of an orthogonal grid. But numerical analysis techniques for processing the collected observations usually require data on an orthogonal grid. Even when data from unevenly spaced points are interpolated to other unevenly spaced points, many techniques require an intermediate orthogonal grid.

In this note, we describe a technique by which data recorded at unevenly spaced sites are used with cubic splines<sup>1</sup> to interpolate to unevenly spaced sites or to generate an orthogonal grid. The technique may be applied to any type of data. Judicious choice of grid spacing will, however, improve performance when the general variation of the specific data being interpolated is known.

### PROCEDURE FOR INTERPOLATING

This procedure for interpolating from unevenly spaced data has three steps. First, we use the known data to generate an initial "guess field" grid by means of a simple square-of-distance weighting scheme using straight-line distance between points. Then, we adjust the initial guess field rows and columns using the known data points and a parabolic leapfrogging technique. Finally, we interpolate the corrected field to a desired orthogonal grid or to another set of unevenly spaced points.

Initial guess field rows and columns are chosen such that each row and column passes through one of the known data points. Normally, only one column is generated per unique data point x value and only one row is generated per unique data point y value. The grid produced is orthogonal but has unevenly spaced rows and columns. After the initial rows and columns are adjusted, cubic spline interpolation<sup>1</sup> is performed

Guardado, Julio L., and William T. Sommers

1977. Interpolation of unevenly spaced data using a parabolic leapfrog correction method and cubic splines. USDA Forest Serv. Res. Note PSW-324, 5 p., Pacific Southwest Forest and Range Exp. Stn., Berkeley, Calif.

The technique proposed allows interpolation of data recorded at unevenly spaced sites to a regular grid or to other sites. Known data are interpolated to an initial guess field grid of unevenly spaced rows and columns by a simple distance weighting procedure. The initial guess field is then adjusted by using a parabolic leapfrog correction and the known data. The final output is then generated by use of cubic spline interpolation on the adjusted grid. Application to test and actual data showed acceptably low root-mean-square error.

*Oxford:* 111.0-015.5.

*Retrieval Terms:* meteorological analysis; data processing; cubic splines.

independently on each row and column to generate two data values at the desired output points. These two data values are then averaged to produce the final output.

### Inverse Square-of-Distance Weighting

The problem of determining a function value at a random point on a grid was solved using an inverse square-of-distance weighting applied to the known point function values. The basic procedure for finding a value at an unknown point from weighted values at  $n$  known points is

$$f' = \frac{\sum_{i=1}^n w_i f_i}{\sum_{i=1}^n w_i}$$

where  $f'$  is the function value at the unknown point,  $f_i$  are the function values at the  $n$  known points, and  $w_i$  are the weights given to the  $f_i$ . Because we desire that the contributions of the  $f_i$  to  $f'$  be weighted as the inverse square of the distance, then

$$w_i = \frac{1}{d_i^2}$$

where  $d_i$  are the distances between known points and unknown point. This procedure is repeated for each unknown point on the row or column of the initial guess grid which we are filling.

For example, say that the point  $x_1$  was a distance  $d_1$  from a point  $x'$  at which we would like to approximate the function value. Furthermore, the point  $x_2$  is twice as far from  $x'$  as  $x_1$ . Point  $x_1$  would be given a weight of  $4/5$  and point  $x_2$  would receive a weight of  $1/5$  because it is twice as far away as  $x_1$  and should be given  $1/4$  the weight of  $x_1$ . The sum of the weights multiplied by the function values at the known points yields the function value at the unknown point.

### The Parabolic Leapfrog

Since all the initial guess rows and columns pass through at least one known data point, the question "How can we correct the initial guess given a known point?" arises. It was decided to correct the initial rows and columns by a "leapfrog" technique beginning at the known data point and performing piecewise corrections using a parabola passed through three points to obtain a fourth point within that interval. The leapfrogging procedure is diagrammed for a case where (a) leftmost endpoint is known, (b) rightmost endpoint is known, and (c) a "central" point is known. The points marked with an asterisk

are used to determine those marked with a question mark. The numbers signify the leapfrogging sequence.

- (a) (+) + + + +  
 1 \* \* ? \*  
 2 \* \* ? \*
- (b) + + + + (+)  
 \* ? \* \* 1  
 \* ? \* \* 2
- (c) + + (+) + +  
 \* \* \* ? \* 1  
 \* ? \* \* 2

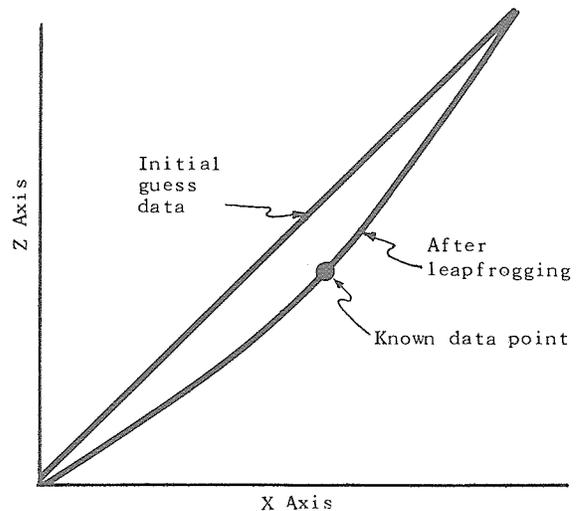


Figure 1—The curve demonstrates the effect of parabolic leapfrogging applied to a straight line and a known data point not on that line.

Figure 1 shows the effect of the leapfrogging procedure on a straight line, given a known point not on the line. We can see by the distance between the two lines that the effect of the correction procedure diminishes as the distance from the known point increases.

Given a set of points

$$(x_i, f_i) \text{ for } i = 0, 1, 2, 3, \dots, n \quad n \geq 3$$

and given a known point  $(x_k, f_k)$   $0 \leq k \leq n$ , then the remaining  $f_i$  ( $i = 0, \dots, n \quad i \neq k$ ) may be corrected by using the technique of parabolic leapfrogging.

Let  $j$  be the starting subscript for choosing the points to be used for interpolation and correction in a negative (decreasing  $x_i$ ) direction, and let  $m$  be the starting subscript for interpolation and correction in a positive (increasing  $x_i$ ) direction. We must consider three cases when choosing  $j$  and  $m$ .

Case 1:  $x_k = x_n$ , then  $j = n$  and  $m = \theta$  (not defined)

Case 2:  $x_k = x_0$ , then  $j = 0$  and  $m = 0$

Case 3:  $x_k \neq 0, n$ , then  $j = \min(n, k+1)$  and  $m = \max(0, k-1)$

We perform the parabolic leapfrog corrections in a positive direction by choosing the set of points

$$\{(x_i, f_i), (x_{i+1}, f_{i+1}), (x_{i+3}, f_{i+3})\}$$

where  $m+3 \leq n$  and  $m \leq i \leq n-3$ , to find the point  $(x_{i+2}, f_{i+2})$ .

For leapfrogging in the negative direction we choose the set of points

$$\{(x_{i-3}, f_{i-3}), (x_{i-1}, f_{i-1}), (x_i, f_i)\}$$

where  $j \geq 3$  and  $j \geq i \geq 3$ , to find the point  $(x_{i-2}, f_{i-2})$ .

We note from the above that at least four points are required for this correction procedure and that the leapfrogging cannot always be done in both a positive and negative direction. Leapfrogging could not be used to correct a boundary value but can be used on interior points nearest to the boundaries. Because interpolation, by definition, requires known data to bound unknown points, correcting boundary values should not be required.

### The Lagrange Interpolating Polynomial

A parabola was fitted through the set of "known" points in the leapfrogging stage using a procedure known as Lagrange interpolation. This technique was chosen because of its ease of use and its flexibility. A polynomial of degree  $n-1$  is fitted through  $n$  points, so that if desired, one could include more points in the leapfrog's individual interpolations with minor modifications to the procedure.

Let

$$P(x) = \sum_{i=0}^{n-1} f_i L_i(x)$$

where  $P(x)$  = the interpolated value at  $x$   
 $f_i$  = the value of the function to be approximated at known point  $x_i$   
 $L_i(x)$  = the  $i^{\text{th}}$  Lagrange coefficient at  $x$   
 $x_i$  = the  $i^{\text{th}}$  known  $x$  value  
 $n$  = the number of interpolating points

We define the Lagrange coefficient as

$$L_k(x) = \prod_{\substack{m=0 \\ m \neq k}}^{n-1} \frac{x-x_m}{x_k-x_m}$$

for  $k = 0, 1, \dots, n-1$ .

Henrici<sup>2</sup> gives the motivation for this technique and also a proof.

### Cubic Splines

When a grid has been generated and corrected, cubic spline interpolation is used to supply data values at the desired unknown points. The cubic spline is a cubic polynomial that can be used with data sets of either fixed or variable mesh length. Cubic spline interpolation provides a minimum curvature fit to the data that is continuous through the second derivative and returns exact values at known points. For additional details see Sommers (1976).<sup>1</sup>

### APPLICATION

We first tested the grid generation scheme by reconstructing an existing grid of 51 rows and 51 columns with a data field consisting of a conical surface centered on gridpoint (26,26). The cone's surface formed a 30° angle with the plane of the grid and the data values ranged from 0.0 at (26,26) to slightly over 20.3 at the grid's corner points. The "known" points for the grid interpolation were chosen at random from every row, every other row, and every third row. The root-mean-square errors of the unknown points in each case are summarized in *table 1*. A FORTRAN IV program compiled by the FORTRAN H optimizing compiler and executed on an IBM 360 model 50<sup>3</sup> was used to perform the interpolation from the known points to the desired 51 X 51 grid. Execution times for each run are included in *table 1* to give a feel for percentage change when the number of points is increased or decreased. The ratio of execution times would be fairly constant for most machines.

Table 1—Error and execution time for interpolation on a conical surface to a 51 x 51 grid

Number of known points	Root-mean-square error <sup>1</sup>	Execution time (min)
51	1.5179	1.53
26	2.5315	0.53
17	3.1588	0.46

<sup>1</sup> Root-mean-square errors were determined by taking the values at known points, dropping them out, interpolating to them as if they were unknown, and comparing the results.

As a second test, we selected a set of 71 standard reporting weather stations in the Southeastern United States, held out a subset as the "unknown" points, and interpolated the "known" points to the unknown points by generating a grid, using the same procedure as in the first test case, and then interpolating the gridded data to the unknown points. *Figure 2* shows

the location of the 50 known points and the 21 unknown points. We performed the interpolation procedure on three variables: temperature, sea level temperature, and dewpoint.

During the initial attempt at interpolating the real-world data we discovered that the "shooting" error which sometimes arises when one is fitting curves became quite large and unacceptable. Shooting errors arise when the gradient between two known points is so large that the interpolating function trajectory continues to increase or decrease beyond reasonable limits at an unknown point downstream. The shooting error was corrected two different ways for comparison. The first and most obvious way to correct it was to limit the interpolated value to either the minimum or the maximum of the points being used for interpolation according to whether we undershot or overshot. The second was to fit the point using linear interpolation if shooting occurred. We ran the program both ways for six hourly observations between 1800 and 2300 GMT on April 16, 1976, and computed root-mean-square errors at each of the "unknown" points. Linear interpolation resulted in minimum root-mean-square errors in all cases analyzed. The results are given in *table 2*.

Table 2—Root-mean-square temperature, sea level temperature, and dewpoint errors at interpolated stations for hour observations between 1800 and 2300 GMT on April 16, 1976, using two methods for correcting shooting error

Correction method	Root-mean-square error for interpolated . . .		
	Temperature	Sea level temperature	Dewpoint
Minimum/maximum limiting	2.46	2.51	3.76
Linear interpolation	2.42	2.43	3.38

The points used as the "known" points were chosen at random and the grid used for the interpolation was arbitrarily given 25 rows and columns. This grid size was chosen to hold computer core and cpu requirements down in an effort to speed up the debugging process by increasing the total number of runs per day. A careful analysis of the known data available and a judicious choice of the grid's mesh length based on the user's particular data base should

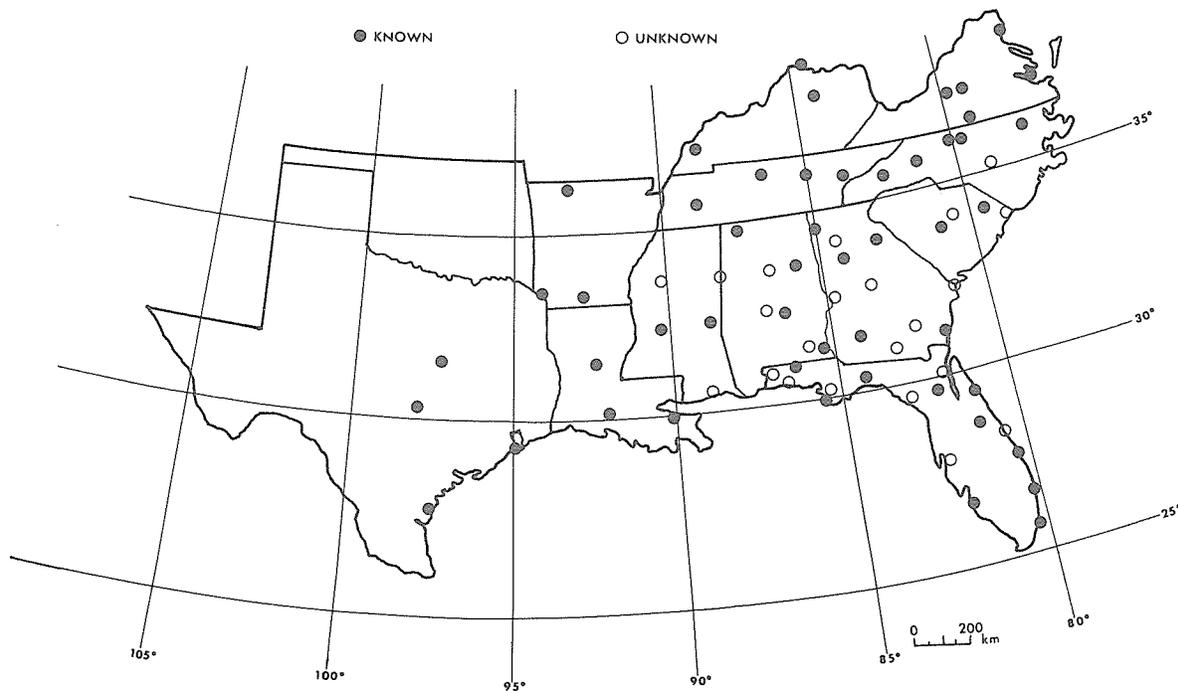


Figure 2—Meteorological data from "known" sites in the 13 southeastern States were interpolated to "unknown" sites in the six most southeasterly States.

yield even better results. These two considerations are the most important when using this particular scheme for interpolation of data.

FORTTRAN IV routines which will perform the interpolation of randomly spaced data to grid, and which will interpolate gridded data to randomly spaced points, are available on request to: Director, Pacific Southwest Forest and Range Experiment Station, P.O. Box 245, Berkeley, California 94701, Attention: Computer Services Librarian.

## NOTES

<sup>1</sup>Sommers, W. T. 1976. *Data interpolation by cubic splines*. USDA Forest Serv. Res. Note PSW-313, 5 p. Pacific Southwest Forest and Range Exp. Stn., Berkeley, Calif.

<sup>2</sup>Henrici, P. 1964. *Elements of numerical analysis*. p. 183-193. John Wiley and Sons, New York.

<sup>3</sup>Trade names and commercial enterprises or products are mentioned solely for information. No endorsement by the U.S. Department of Agriculture is implied.

### The Authors

---

**JULIO L. GUARDADO**, an assistant programmer, performed the work reported here while assigned to the Station's forest fire meteorology research unit, at the Forest Fire Laboratory, Riverside, California. He earned a B.S. degree in applied science at the University of California, Riverside in 1976. He is presently a programmer-analyst with California Computer Products (CalComp), Anaheim, Calif.

**WILLIAM T. SOMMERS**, a research meteorologist, is assigned to the forest fire meteorology research unit, at Riverside, California. He earned degrees in meteorology at City College of New York (B.S., 1965), Massachusetts Institute of Technology (M.S., 1967), and State University of New York at Albany (Ph.D., 1972). He joined the Forest Service and Station staff in 1973.