BILINEAR INTERPOLATION

PURPOSE

Perform a bilinear interpolation of a series of data points.

DESCRIPTION

Two dimensional interpolation takes a series of (x,y,z) points and generates estimated values for z's at new (x,y) points. Interpolation is used when the function that generated the original (x,y,z) points is unknown. Interpolation is related to, but distinct from, fitting a function to a series of points. In particular, an interpolated function goes through all the original points while a fitted function may not.

There are two distinct types of 2d interpolation. In the first, data is available for a rectangular grid of points and interpolation is performed for points off the grid. In the second, data is available for a random set of points and the interpolation is generated on a rectangular grid. This second form can be used to generate a contour or surface plot when the data do not form a grid.

The BILINEAR INTERPOLATION and BIVARIATE INTERPOLATION commands are used for the first type. The bilinear interpolation is analogous to linear interpolation. A weighted average of the 4 surrounding grid points is used to determine the interpolated value. The BIVARIATE INTERPOLATION command uses piecewise polynomials to do the interpolation and is generally more accurate. See the documentation for BIVARIATE INTERPOLATION command for details. The 2D INTERPOLATION command is used for the second type of interpolation.

SYNTAX

LET <z2> = BILINEAR INTERPOLATION <z1> <y1> <x1> <y2> <x2> <SUBSET/EXCEPT/FOR qualification>
where <z1> is a variable containing the z-axis data points;
<y1> is a variable containing the vertical axis data points;
<x1> is a variable containing the horizontal axis data points;
<x2> is a variable containing the horizontal points where the interpolation is to be performed;
<y2> is a variable containing the vertical points where the interpolation is to be performed;
<z2> is a variable (same length as <x2>) where the interpolated values are stored;
and where the <SUBSET/EXCEPT/FOR qualification> is optional.

EXAMPLES

LET Z2 = BILINEAR INTERPOLATION Z1 Y1 X1 Y2 X2

NOTE 1

The <z1>, <y1>, and <x1> arrays should be the same size. In addition, <y1> and <x1> must form a rectangular grid. The rectangular grid must be complete. No provision is made for missing data values on the grid. The <x2> and <y2> arrays must be the same size. An error message is printed if any of these conditions is not met.

NOTE 2

The interpolation points (i.e., <x2> and <y2>) must be within the range of the original data points (i.e., <x1>). An error message is generated if this is not the case.

NOTE 3

The original data do not have to be in sorted order.

DEFAULT

None

SYNONYMS

None

RELATED COMMANDS

INTERPOLATION	=	Compute a cubic spline interpolation of a series of points.
LINEAR INTERPOLATION	=	Compute a linear interpolation of a series of points.
BIVARIATE INTERPOLATION	=	Compute a bivariate interpolation from a grid to random points of a 2D series of points.
2D INTERPOLATION	=	Compute a bivariate interpolation from a 2D series of points to a rectangular grid.
FIT	=	Perform a least squares fit.

REFERENCE

"Numerical Recipes: The Art of Scientific Computing (FORTRAN Version)," Press, Flannery, Teukolsky, and Vetterling, Cambridge University Press, 1989 (chapter 3).

APPLICATIONS

Mathematics

IMPLEMENTATION DATE

94/6

PROGRAM

```
LET X = SEQUENCE -4 1 4 FOR I = 1 1 81
LET Y = SEQUENCE -4 9 1 4
LET Z = X**2 + Y**2 - X*Y
.
READ X2 Y2
-3.5 -2.8
-1.5 -0.8
-0.6 3.2
2.1 -3.1
3.8 3.2
END OF DATA
.
LET Z2 = BILINEAR INTERPOLATION Z Y X Y2 X2
SET WRITE DECIMALS 3
PRINT X2 Y2 Z2
```

The following output is generated.

```
VARIABLES--X2             Y2             Z2

          -3.500         -2.800         10.700
          -1.500         -0.800          2.100
          -0.600          3.200         12.920
           2.100         -3.100         20.710
           3.800          3.200         12.840
```