

## 2D INTERPOLATION

### PURPOSE

Perform a bivariate interpolation of a series of scattered 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 does 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 2D INTERPOLATION command is used for the second type of interpolation. The method DATAPLOT uses is based on the construction of local thin plate splines, which are then blended together via univariate piecewise Hermite cubic polynomials. The method is described in detail in the paper listed in the REFERENCE section below. The BILINEAR INTERPOLATION and BIVARIATE INTERPOLATION commands are used for the first type. See the documentation for these commands for more details.

### SYNTAX

LET <z2> = 2D 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 = 2D INTERPOLATION Z1 Y1 X1 Y2 X2
```

### NOTE 1

The <z1>, <y1>, and <x1> arrays should be the same size, but they do not need to be in any particular order. The <x2> and <y2> define the grid points in the x and y axis and need not be the same size. Only the distinct points in <x2> and <y2> are used. That is, DATAPLOT generates the full grid from the unique points in <x2> and <y2>. If the number of unique points in <x2> is NX and the number of unique points in <y2> is NY, then the returned <z2> array will have NX\*NY points. Both the <x2> and <y2> arrays are sorted from lowest value to highest value. The <z2> points are returned in the grid order. That is, starting with the smallest <x2> value, a value is returned for each <y2> value in sorted order, then for the second smallest <x2> value, and so on. The <x2> and <y2> arrays are not modified on output.

Often you will need the full grid of (x,y) points corresponding to the interpolated points for subsequent contour or 3d plots. This can be easily generated with the SEQUENCE command. For example, suppose your grid goes from -4 to 4 in both the X and Y directions with an increment of 1. Then the grid points can be generated with the following commands:

```
LET X = SEQUENCE -4 1 4 FOR I = 1 1 81
LET Y = SEQUENCE -4 9 1 4
```

See the documentation for the SEQUENCE command in this chapter for more details on this command.

### NOTE 2

The desired average number of points per region can be specified with the following command:

```
LET NPPR = <value>
```

where <value> is a number greater than or equal to 3. The default is 10, which should give good results in most cases.

### NOTE 3

Bivariate interpolation of scattered data can be ill-behaved for some data sets. Given that there can be various types of irregularly spaced data, there have been various approaches to this problem. The best algorithm is probably dependent on the particular type of data

that you have. DATAPLOT uses the LOTPS library written by Richard Franke of the Naval Postgraduate School. Although this algorithm should work well with many data sets, there may be some where it does not. For this reason, it is a good idea to examine the output for this command carefully to see that reasonable results are obtained.

**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 bilinear interpolation from a grid to random points of a 2D series of points.
BILINEAR INTERPOLATION	=	Compute a bilinear interpolation from a grid to random points of a 2D series of points.
CONTOUR PLOT	=	Generates a contour plot.
3D-PLOT	=	Generates a 3d function or data plot.
FIT	=	Perform a least squares fit.

**REFERENCES**

“Smooth Interpolation of Scattered Data by Local Thin Plate Splines,” Richard Franke, Naval Postgraduate School, TR#NPS-53-81-002, 1981 (also available from NTIS as AD-A098 232/2).

**APPLICATIONS**

Mathematics

**IMPLEMENTATION DATE**

94/6

**PROGRAM**

```
LET X = DATA 0 0.2 0.5 0.6 1.0
LET Y = DATA 0.2 0.7 0 1.0 0.5
LET Z = DATA 0.2 0.5 0.6 0.3 0
LET X2 = DATA 0.2 0.9
LET Y2 = DATA 0.1 0.8
.
SET WRITE DECIMALS 3
LET Z2 = 2D INTERPOLATION Z Y X Y2 X2
PRINT Z2
```

The following output is generated.

VARIABLES--Z2

```
0.378
0.336
0.506
0.057
```