CHAPTER

30

The G3GRID Procedure

Overview

The G3GRID procedure processes an existing SAS data set to create a data set that the G3D or GCONTOUR procedure can use to produce three-dimensional surface or contour plots. The procedure creates a data set whose horizontal (*x* and *y*) variable values form a complete grid, and it interpolates the value of the vertical (*z*) variables for each point on the *x-y* plane. Using the G3GRID procedure, you can

- [ ] create a rectangular grid of interpolated or smoothed values from irregularly spaced observations for use in a three-dimensional surface or contour plot

- [ ] complete a rectangular grid of interpolated or smoothed values for an input data set that has an insufficient number of observations to produce a three-dimensional surface or contour plot

- [ ] interpolate or smooth noisy data for a three-dimensional graph.

The G3GRID procedure does not produce graphics output. Instead, it produces an output data set that you can use as the input data set for the G3D or GCONTOUR procedure.

Figure 30.1 on page 1008 and Figure 30.2 on page 1008 illustrate the effect of the G3GRID procedure on data.

Figure 30.1 on page 1008 shows a collection of data points, where *z=f*(*x, y*). These points are randomly distributed and cannot be displayed with a G3D surface plot, although they can be displayed with a scatter plot.

Figure 30.1 Scatter Plot of Data Set before G3GRID Processing (GR30N01(a))
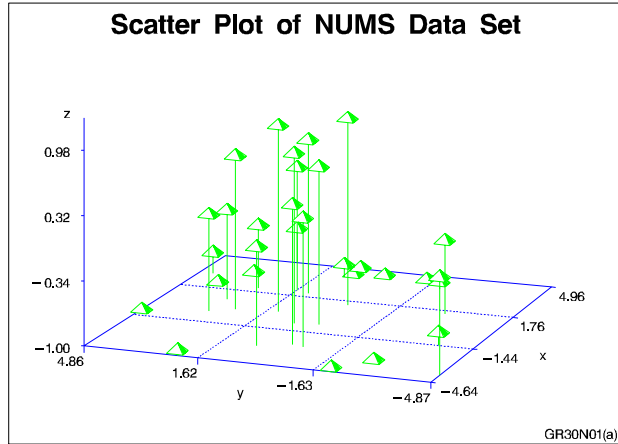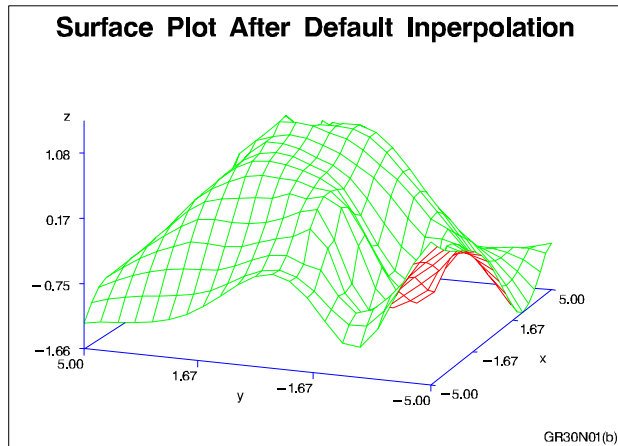

Figure 30.2 on page 1008 shows a surface plot of the data set that is created by a G3GRID interpolation of the original data set shown in Figure 30.1 on page 1008.

Figure 30.2 Surface Plot of Data Set after G3GRID Processing (GR30N01(b))

Note: The evenly distributed horizontal (*x, y*) data points form a grid for the three-dimensional graph. △

Concepts

About the Input Data Set

The input data set must contain at least three numeric variables:

□ two horizontal variables, (*x, y*)

□ one or more vertical variables, *z* through *z-n*, that will be interpolated or smoothed as if it were a function of the two horizontal variables.

The procedure can process multiple vertical variables for each pair of horizontal variables that you specify. If you specify more than one vertical variable, the G3GRID procedure performs a separate analysis and produces interpolated or smoothed values for each vertical variable. If more than one observation in the input data set has the same values for both horizontal variables, *x* and *y*, a warning message is printed, and only the first such point is used in the interpolation.

By default, the interpolation is performed after both variables are similarly scaled because the interpolation methods assume that the scales of *x* and *y* are comparable.

Multiple Vertical Variables

In the GRID statement, you can name multiple vertical variables (*z* through *z-n*) and produce a data set that contains two horizontal variables and multiple vertical variables. You can use the resulting data set to produce plots of the relationships of the two horizontal variables to different vertical variables.

Horizontal Variables Along a Nonlinear Curve

If the points that are generated by the horizontal variables tend to lie along a curve, a poor interpolation or spline may result. In such cases, the vertical variable(s) and one of the horizontal variables should be modeled as a function of the remaining horizontal variable. You can use a scatter plot of the two horizontal variables to help determine the appropriate function.

If the horizontal variable points are collinear, the procedure interpolates the function as constant along lines perpendicular to the line in the plane that is generated by the input data points.

About the Output Data Set

The output data set contains the two horizontal variables, the interpolated or smoothed vertical variables, and the BY variables, if any. If the GRID statement's SMOOTH= option is used, the output data set also contains a variable named _SMTH_, with a value equal to that of the smoothing parameter.

You can control both the number of *x* and *y* values in the output data set and the values themselves. In addition, you can specify an interpolation method.

Interpolation Methods

The G3GRID procedure can use one of three interpolation methods: bivariate interpolation (the default), spline interpolation, and smoothed spline interpolation.

Default Bivariate Interpolation

Unless you specify the SPLINE option, the G3GRID procedure is an interpolation procedure. That is, it calculates z values for x, y points that are missing from the input grid. The surface that is formed by the interpolated data passes precisely through the data points in the input data set.

This default method of interpolation works best for fairly smooth functions with values given at uniformly distributed points in the plane. If the data points in the input data set are erratic, the default interpolated surface can be erratic.

This default method is a modification of that described by Akima (1978). This method consists of

1 dividing the plane into nonoverlapping triangles that use the positions of the available points

2 fitting a bivariate fifth degree polynomial within each triangle

3 calculating the interpolated values by evaluating the polynomial at each grid point that falls in the triangle.

The coefficients for the polynomial are computed based on

□ the values of the function at the vertices of the triangle

□ the estimated values for the first and second derivatives of the function at the vertices.

The estimates of the first and second derivatives are computed using the n nearest neighbors of the point, where n is the number specified in the GRID statement's NEAR= option. A Delauney triangulation (Ripley 1981, p. 38) is used for the default method. The coordinates of the triangles are available in an output data set if requested by the OUTTRI= option in the PROC G3GRID statement.

Spline Interpolation

If you specify the SPLINE option, a method is used that produces an interpolation or smoothing that is optimally smooth in a certain sense (Harder and Desmarais 1972, Meinguet 1979). The surface that is generated can be thought of as one that would be formed if a stiff, thin metal plate were forced through or near the given data points. For large data sets, this method is substantially more expensive than the default method.

The function u, formed when you specify the SPLINE option, is determined by letting

$$t_j = (x_j, y_j)$$

$$t = (x, y)$$

and

$$|t - t_j| = \left((x - x_j)^2 + (y - y_j)^2 \right)^{1/2}$$

$$\mathbf{u}(x, y) = \Sigma_{j=1}^n c_j \mathrm{E}(t, t_j) + d_0 + d_1 x + d_2 y$$

where

$$E\left(s, t\right) = |s - t| \, log\left(|s - t|\right).$$

The coefficients c_1, c_2,..., c_n and d_1, d_2, d_3 of this polynomial are determined by these equations:

$$\left(\mathbf{E} + \mathbf{n}\lambda\mathbf{I}\right)\mathbf{c} + \mathbf{T}\,\mathbf{d} = \mathbf{z}$$

and

$$\mathbf{T}'\mathbf{c} = \mathbf{0}$$

where

E
   is the $n \times n$ matrix $E(t_i\,,\ t_j\,)$

I
   is the $n \times n$ identity matrix

λ
   is the smoothing parameter that is specified in the SMOOTH= option

c
   is $(c_1\,,...,\ c_n\,)$

z
   is $(z_1\,,...,\ z_n\,)$

d
   is $(d_1,\ d_2,\ d_3)$

T
   is the $n \times 3$ matrix whose *i*th row is $(1,\ x_i,\ y_i)$.

See Wahba (1979) for more detail.

Spline Smoothing

To produce a smoothed spline, you can use the GRID statement's SMOOTH= option with the SPLINE option. The value or values specified in the SMOOTH= option are substituted for λ in the equation that is described in "Spline Interpolation" on page 1010. A smoothed spline trades closeness to the original data points for smoothness. To find a value that produces the best balance between smoothness and fit to the original data, you can try several values for the SMOOTH= option.

Procedure Syntax

Requirements: Exactly one GRID statement is required.

Reminder: The procedure can include the SAS/GRAPH BY statement.

Supports: Output Delivery System (ODS)

> **PROC G3GRID** <DATA=*input-data-set*>
>   <OUT=*output-data-set*>
>   <OUTTRI=*output-data-set*>;
> **GRID** *grid-request* </option(s)>;

PROC G3GRID Statement

Identifies the input data set. Optionally specifies one or two output data sets.

Requirements: An input data set is required.

Syntax

> **PROC G3GRID** <DATA=*input-data-set*>
>   <OUT=*output-data-set*>
>   <OUTTRI=*output-data-set*>;

Options

DATA=*input-data-set*
    specifies the SAS data set that contains the variables to process. By default, the
    procedure uses the most recently created SAS data set.
    See also: "SAS Data Sets" on page 25 and "About the Input Data Set" on page 1009

OUT=*output-data-set*
    specifies the output data set. The data set contains any BY variables that you
    specify, the interpolated or smoothed values of the vertical variables (z through z-n),
    and the coordinates for all grid positions on the horizontal (x-y) plane. If you specify
    smoothing, the output data set also contains a variable named _SMTH_, whose value
    is a smoothing parameter. The observations in this data set are ordered by any
    variables that you specify with a BY statement. By default, the output of PROC
    G3GRID creates WORK.DATA1.
        Depending on the shape of the original data and the options you use, the output
    data set may contain values for the vertical (z through z-n) values that are outside of
    the range of the original values in the data set.
    Featured in: Example 1 on page 1016

OUTTRI=*output-data-set*
    specifies an additional output data set that contains triangular coordinates. The data
    set will contain any BY variables that you specify, the two horizontal variables giving
    the horizontal (x -y) plane coordinates of the input points, and a variable named
    TRIANGLE that uses integer values to label the triangles. The observations in this
    data set are ordered by any variables that you specify with a BY.
        The data set contains three observations for each value of the variable
    TRIANGLE. The three observations give the coordinates of the three vertices of the
    triangle. Points on the convex hull of the input data set of points are also assumed to
    lie in degenerate triangles whose other vertices are at infinity. The points in the
    convex hull can be recovered by keeping only those triangles with exactly two
    missing vertices.
        By default, no OUTTRI= data set is produced. OUTTRI= is not valid when you
    specify the SPLINE option in the GRID statement.

GRID Statement

Specifies the three numeric variables for interpolation or smoothing. Optionally specifies the number of observations (*x* and *y* values) in the output data set; output values for the two horizontal variables *x,y*; and the interpolation method for the vertical variables.

Requirements: Exactly one grid request is required.

Syntax

GRID *grid-request </option(s)>*;

grid-request must be:

   y*x=z(s)

 grid-request must be

y*x=z(s)

option(s) can be one or more options from any or all of the following categories:

☐ grid options:

    AXIS1=*ascending-value-list*

    AXIS2=*ascending-value-list*

    NAXIS1=*n*

    NAXIS2=*n*

☐ interpolation options:

    JOIN

    NEAR=*n*

    NOSCALE

    PARTIAL

    SMOOTH=*ascending-value-list*

    SPLINE

Required Arguments

y*x=z(s)
   specifies three or more numeric variables from the input data set:

y
   is one of the variables that forms the horizontal (*x-y*) plane.

x
   is another of the variables that forms the horizontal (*x-y*) plane.

z(s)
   is one or more vertical variables for the interpolation.
   Although the GRID statement can specify only two horizontal variables, it can include multiple vertical variables. Separate vertical variables with blanks:

```
grid x*y=z w u v;
```

Options

AXIS1=*ascending-value-list*
specifies a list of numeric values to assign to the first (*y*) variable in the grid request for the output data set. Numbers that you specify with this option determine the number of values for *y* and override a value that you specify with the NAXIS1= option. *Ascending-value-list* must be in ascending order. It is either an explicit list of values, or a starting and an ending value with an interval increment, or a combination of both forms:

    n <...n>

    n TO *n* <BY *increment*>

    n <...n> TO *n* <BY *increment* > <*n <...n>* >

Featured in: Example 1 on page 1016 and Example 4 on page 1022

AXIS2=*ascending-value-list*
specifies a list of numeric values to assign to the second (*x*) variable in the grid request for the output data set. Numbers that you specify with this option determine the number of values for *x* and override a value that you specify with the NAXIS2= option. *Ascending-value-list* must be in ascending order. It is either an explicit list of values, or a starting and an ending value with an interval increment, or a combination of both forms:

    n <...n>

    n TO *n* <BY *increment*>

    n <...n> TO *n* <BY *increment* > <*n <...n>* >

Featured in: Example 1 on page 1016 and Example 4 on page 1022

JOIN
uses a linear interpolation within a set of triangular regions that are formed from the input data set. This interpolation method creates values in the range of the initial values of the vertical variable, but the resulting interpolated surface may not be smooth.

NAXIS1=*n*
specifies the number of values for the first (*y*) variable in the grid request for the output data set. You can determine the actual values used for *y* by taking the minimum and maximum values of *y* and dividing the range into *n*–1 equal sections. By default, NAXIS1=11.
   A value specified with NAXIS1= is ignored if values are also specified with AXIS1=.

NAXIS2=*n*
specifies the number of values for the second (*x*) variable in the grid request for the output data set. You can determine the actual values that are used for *x* by taking the minimum and maximum values of *x* and dividing the range into *n*–1 equal sections. By default, NAXIS2=11.
   A value specified with NAXIS2= is ignored if values are also specified with AXIS2=.

NEAR=*n*
specifies the number of nearest data points to use for computing the estimates of the first and second derivatives. As NEAR= values become larger, time and computation costs increase significantly. NEAR= is ignored if you specify SPLINE. The value of *n* must be greater than or equal to 3. By default, NEAR=3.
   If the number of input data points is insufficient for the number that you specify with NEAR=, a smaller number of data points is used.

Featured in: Example 3 on page 1021

NOSCALE
specifies that the *x* and *y* variables not be scaled to the same range before interpolation. By default, the interpolation is performed after both variables are similarly scaled because the interpolation methods assume that the scales of *x* and *y* are comparable.

PARTIAL
specifies that a spline be used to estimate the derivatives for the biquintic polynomial interpolation. A bivariate spline is fit to the nearest neighbors and used to estimate the needed derivatives. This option produces results that are less smooth than those produced by the SPLINE option and uses fewer computer resources. However, the results produced by PARTIAL are smoother than those that are produced by the default. If you use both PARTIAL and the SPLINE option, PARTIAL is ignored.

Featured in: Example 3 on page 1021

SMOOTH=*ascending-value-list*
specifies a list of numbers for smoothing parameters. Use this option only when you also use the SPLINE option. *Ascending-value-list* must be in ascending order. It is either an explicit list of values, or a starting and an ending value with an interval increment, or a combination of both forms:

n <...*n*>

n TO *n* <BY *increment*>

n <...*n*> TO *n* <BY *increment* > <*n* <...*n*> >

For each value λ of the smoothing parameter, a function *u* (*x, y*) is formed that minimizes

$$\frac{1}{n}\Sigma_{j=1}^{n}\left(\mathbf{u}\left(\mathbf{x_j}, \mathbf{y_j}\right) - \mathbf{z_j}\right)^2 + \lambda\Sigma_{j=0}^{2} \int\limits_{-\infty}^{\infty} \int^{\infty 2} dxdy$$

where *n* is the number of data points and the pairs (*x_j*, *y_j*) are the available points, with corresponding function values *z_j* (Wahba 1979).

The higher the value of the smoothing parameter, the smoother the resulting interpolation. The lower the smoothing parameter, the closer the resulting surface is to the original data points. A smoothing parameter of 0 produces the same results as the SPLINE option without SMOOTH=.

This procedure repeats for each value of the smoothing parameter. The output data set that you specify in the OUT= option contains the interpolated values, the values of the grid points, and the values of the smoothing parameter in the variable _SMTH_. The output data set contains a separate grid for each value of the smoothing parameter.

Featured in: Example 2 on page 1019

SPLINE
specifies the use of a bivariate spline (Harder and Desmarais 1972, Meinguet 1979) to interpolate or to form a smoothed estimate if you also use the SMOOTH= option. This option results in the use of an order n^3 algorithm, where *n* is the number of input data points. Consequently, this method can be time-consuming. If you use more than 100 input points, the procedure may use excessive time.

Featured in: Example 2 on page 1019 and Example 4 on page 1022

Controlling Observations in the Output Data Set

By default, the G3GRID procedure produces a data set with 121 observations for combinations of 11 values for each of the horizontal variables, *x* and *y*. To create a data

set with a different number of observations, use the GRID statement's NAXIS1= or NAXIS2= options to specify the number of the values of *y* or *x*, respectively. Or, use the GRID statement's AXIS1= or AXIS2= options to specify the actual values for *y* or *x*, respectively.

Table 30.1 on page 1016 shows the number of observations that will be in the output data set if you use any of these options.

Table 30.1   Number of Observations Contained in the Output Data Set

Options Specified	Number of Observations in Output Data Set
None	121
AXIS1=	(number of values for AXIS1=) * 11
AXIS2=	(number of values for AXIS2=) * 11
NAXIS1=	(value of NAXIS1=) * 11
NAXIS2=	(value of NAXIS2=) * 11
AXIS1=, AXIS2=	(number of values for AXIS1=) * (number of values for AXIS2=)
AXIS1=, NAXIS1=	(number of values for AXIS1=) * 11
AXIS1=, NAXIS2=	(number of values for AXIS1=) * (value of NAXIS2=)
AXIS2=, NAXIS1=	(number of values for AXIS2=) * (value of NAXIS1=)
AXIS2=, NAXIS2=	(number of values for AXIS2=) * 11
NAXIS1=, NAXIS2=	(value of NAXIS1=) * (value of NAXIS2=)

If you specify multiple smoothing parameters, the number of observations in the output data set will be the number shown in Table 30.1 on page 1016 multiplied by the number of smoothing values that you specify in the SMOOTH= option. If you use BY-group processing, multiply the number in the table by the number of BY groups.

Depending on the shape of the original data and the options that you specify, the output data set may contain values for the vertical (*z*) values that are outside of the range of the original values in the data set.

Examples

Example 1: Using the Default Interpolation Method

Procedure features:
    G3GRID statement options:
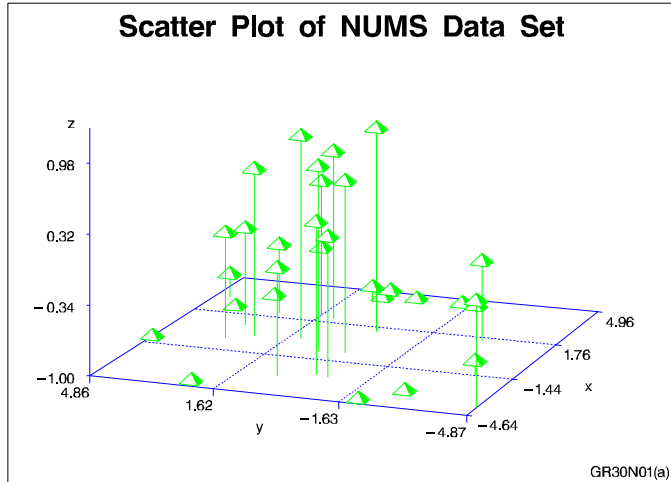   OUT=
    GRID statement options:
        AXIS1=
        AXIS2=
**Other features:**
    DATA step

G3D procedure

Sample library member: GR30N01

Scatter Plot of NUMS Data Set

GR30N01(a)

This example demonstrates the default interpolation method that is used by the GRID statement. The example first generates a scatter plot of random data to show the concentration of data values before processing with the G3GRID procedure. The original data do not contain enough combinations of *x*, *y*, and *z* values to generate a surface plot with the G3D procedure, or a contour plot with the GCONTOUR procedure.

The example then runs the G3GRID procedure to interpolate additional *x*, *y*, and *z* values. Because no interpolation method is specified, the default interpolation method is used. The resulting output data set is used as input to the G3D procedure, which generates the surface plot shown in the following output.
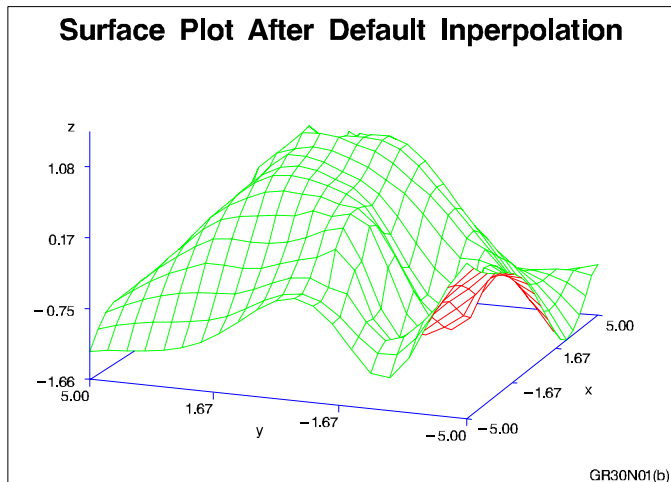
Surface Plot After Default Inperpolation

GR30N01(b)

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         colors=(black blue green red)
         ftext=swiss ftitle=swissb htitle=6 htext=3;
```

Create data set. REFLIB.NUMS uses a set of randomly sampled points to create the data used in this and all remaining examples in this chapter.

```
data reflib.nums;
   keep x y z;
   do i=1 to 30;
      x=10*ranuni(33)-5;
      y=10*ranuni(35)-5;
      z=sin(sqrt(x*x+y*y));
      output;
   end;
run;
```

Define title and footnote.

```
title 'Scatter Plot of NUMS Data Set';
footnote j=r 'GR30N01(a) ';
```

Generate the scatter plot.

```
proc g3d data=reflib.nums;
   scatter y*x=z;
run;
```

Process points with PROC G3GRID. OUT= on G3GRID specifies a name for a temporary output data set. GRID specifies the variables Y*X=Z for the output data set. AXIS@@@ 1

```
proc g3grid data=reflib.nums out=default;
   grid y*x=z / axis1=-5 to 5 by .5
                axis2=-5 to 5 by .5;
run;
```

Define new title and footnote.

```
title 'Surface Plot after Default Interpolation';
footnote j=r 'GR30N01(b) ';
```

Generate a surface plot. The G3D procedure uses as its input data set the G3GRID procedure's output data set.

```
proc g3d data=default;
   plot y*x=z;
```

```
run;
quit;
```

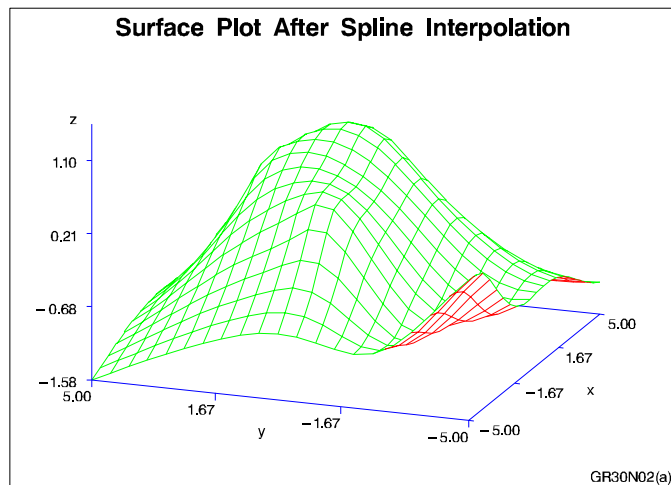# Example 2: Using Spline Interpolation and a Smoothed Spline

Procedure features:

GRID statement options:

SMOOTH=
SPLINE

Data set: REFLIB.NUMS on page 1018

Sample library member: GR30N02

This example extends Example 1 to specify a spline interpolation method on the GRID statement. The output data set, when used in PROC G3D, generates a smoother surface plot than the surface plot that results from the default interpolation shown in Example 1.

This example then specifies a smoothed spline interpolation method on the GRID statement. As shown by the following output, the resulting surface plot is smoother still.
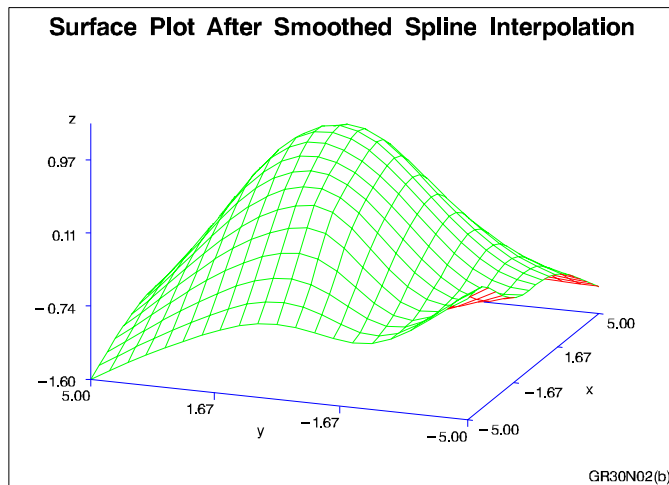
Surface Plot After Smoothed Spline Interpolation

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         colors=(black blue green red)
         ftext=swiss ftitle=swissb htitle=5 htext=3;
```

Define title and footnote.

```
title 'Surface Plot After Spline Interpolation';
footnote j=r 'GR30N02(a) ';
```

Process points with PROC G3GRID. SPLINE specifies the bivariate spline method for the data set interpolation.

```
proc g3grid data=reflib.nums out=spline;
   grid y*x=z / spline
                axis1=-5 to 5 by .5
                axis2=-5 to 5 by .5;
run;
```

Generate a surface plot.

```
proc g3d data=spline;
   plot y*x=z ;
run;
```

Define title and footnote for second plot.

```
title 'Surface Plot After Smoothed Spline Interpolation';
footnote j=r 'GR30N02(b) ';
```

Process points with PROC G3GRID. SMOOTH= specifies the smoothing parameter to use during spline interpolation.

```
proc g3grid data=reflib.nums out=smoothed;
   grid y*x=z / spline
                smooth=.05
                axis1=-5 to 5 by .5
                axis2=-5 to 5 by .5;
run;
```

Generate a surface plot.

```
proc g3d data=smoothed;
   plot y*x=z;
run;
quit;
```

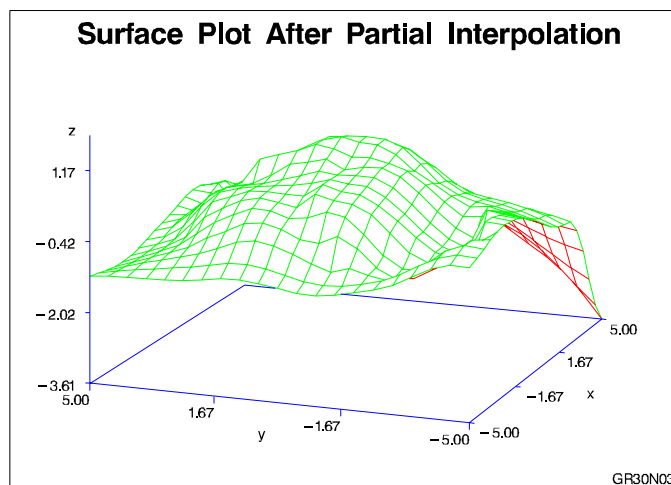# Example 3: Using Partial Spline Interpolation

Procedure features:
  GRID statement options:
    NEAR
    PARTIAL
Data set: REFLIB.NUMS on page 1018
Sample library member: GR30N03

This example specifies a partial spline interpolation method on the GRID statement, using eight nearest neighbors for computing the estimates of the first and second

derivatives. The output data set, when used in PROC G3D, generates a smoother
surface plot than the surface plot that results from the default interpolation shown in
Example 1, but not as smooth as the surface plot that results from the spline
interpolation shown in Example 2.

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         colors=(black blue green red)
         ftext=swiss ftitle=swissb htitle=6 htext=3;
```

Define title and footnote.

```
title 'Surface Plot after Partial Interpolation';
footnote j=r 'GR30N03 ';
```

Process points with PROC G3GRID. PARTIAL specifies that a spline be used to estimate the
derivatives for the biquintic polynomial interpolation. NEAR= specifies the number of nearest
neighbors to be used for computing the estimates of the first and second derivatives.

```
proc g3grid data=reflib.nums out=partial;
   grid y*x=z / partial
                near=8
                axis1=-5 to 5 by .5
                axis2=-5 to 5 by .5;
run;
```

Generate the surface plot.

```
proc g3d data=partial;
   plot y*x=z;
run;
quit;
```

Example 4: Using Spline Interpolation
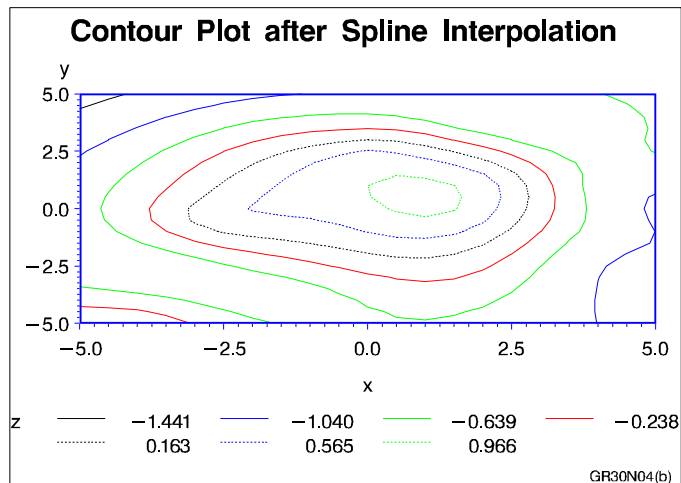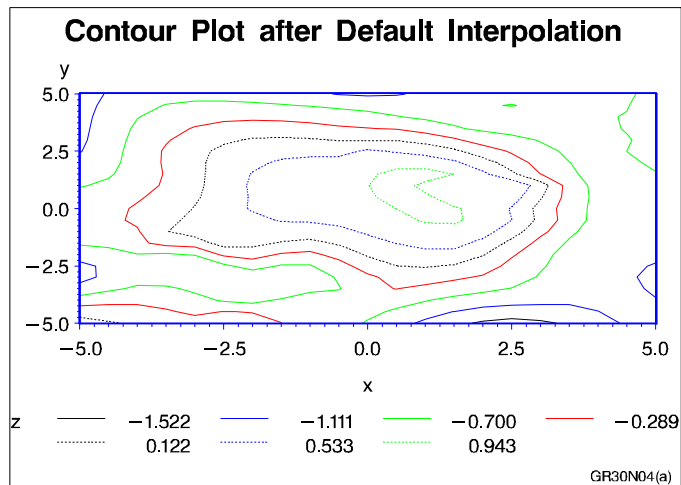
Procedure features:
   GRID statement options:
      AXIS1=
      AXIS2=
      SPLINE
Data set: REFLIB.NUMS on page 1018
Sample library member: GR30N04

   This example demonstrates the default and spline interpolation methods when used by the GCONTOUR procedure to generate contour plots from the resulting output data sets.

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         colors=(black blue green red)
         ftext=swiss ftitle=swissb htitle=6 htext=3;
```

Define title and footnote.

```
title 'Contour Plot after Default Interpolation';
footnote j=r 'GR30N04(a) ';
```

Define axis characteristics.

```
axis1 width=3;
```

Process points with PROC G3GRID. AXIS@@@ 1

```
proc g3grid data=reflib.nums out=numdef;
   grid y*x=z / axis1=-5 to 5 by .5
                axis2=-5 to 5 by .5;
run;
```

Generate the contour after default interpolation.

```
proc gcontour data=numdef;
   plot y*x=z / haxis=axis1 vaxis=axis1;
run;
```

Define new title and footnote.

```
title 'Contour Plot after Spline Interpolation';
footnote j=r 'GR30N04(b) ';
```

Process points with PROC G3GRID. SPLINE specifies the bivariate spline method for the data set interpolation.

```
proc g3grid data=reflib.nums out=numspl;
   grid y*x=z / spline
                axis1=-5 to 5 by .5
                axis2=-5 to 5 by .5;
run;
```

Show the contour after spline interpolation.

```
proc gcontour data=numspl;
   plot y*x=z / haxis=axis1 vaxiss=axis1;
run;
quit;
```

References

Akima, Hiroshi (1978), "A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points," *ACM Transaction on Mathematical Software*, 4, 148–159.

Harder, R.L. and Desmarais, R.N. (1972), "Interpolation Using Surface Splines," *Journal of Aircraft*, 9, 189–191.

Meinguet, Jean (1979), "Multivariate Interpolation at Arbitrary Points Made Simple," *Journal of Applied Mathematics and Physics*, 30, 292–304.

Ripley, B.D. (1981), *Spatial Statistics*, New York: John Wiley & Sons, Inc.

Wahba, Grace (1979), "How to Smooth Curves and Surfaces with Splines and Cross-validation," in U.S. Army Research Office Report 79–2, *Proceedings of the 24th Conference on the Design of Experiments*.