Overview: GAMPL Procedure

The GAMPL procedure is a high-performance procedure that fits generalized additive models that are based on low-rank regression splines (Wood 2006). This procedure provides powerful tools for nonparametric regression and smoothing.

Generalized additive models are extensions of generalized linear models. They relax the linearity assumption in generalized linear models by allowing spline terms in order to characterize nonlinear dependency structures. Each spline term is constructed by the thin-plate regression spline technique (Wood 2003). A roughness penalty is applied to each spline term by a smoothing parameter that controls the balance between goodness of fit and the roughness of the spline curve. PROC GAMPL fits models for standard distributions in the exponential family, such as normal, Poisson, and gamma distributions.

PROC GAMPL runs in either single-machine mode or distributed mode.

NOTE: Distributed mode requires SAS High-Performance Statistics.

PROC GAMPL Features

PROC GAMPL offers the following basic features:

- estimates the regression parameters of a generalized additive model that has fixed smoothing parameters by using penalized likelihood estimation
- estimates the smoothing parameters of a generalized additive model by using either the performance iteration method or the outer iteration method
- estimates the regression parameters of a generalized linear model by using maximum likelihood techniques
- tests the total contribution of each spline term based on the Wald statistic
- provides model-building syntax in the CLASS statement and effect-based parametric effects in the MODEL statement, which are used in other SAS/STAT analytic procedures (in particular, the GLM, LOGISTIC, GLIMMIX, and MIXED procedures)
- provides response-variable options
- enables you to construct a spline term by using multiple variables
- provides control options for constructing a spline term, such as fixed degrees of freedom, initial smoothing parameter, fixed smoothing parameter, smoothing parameter search range, user-supplied knot values, and so on
- provides multiple link functions for any distribution
- provides a WEIGHT statement for weighted analysis
- provides a FREQ statement for grouped analysis
PROC GAMPL Contrasted with PROC GAM

Both the GAMPL procedure and the GAM procedure in SAS/STAT software fit generalized additive models. However, the GAMPL procedure uses different approaches for constructing spline basis expansions, fitting generalized additive models, and testing smoothing components. The GAMPL procedure focuses on automatic smoothing parameter selection by using global model-evaluation criteria to find optimal models. The GAM procedure focuses on constructing models by fitting partial residuals against each smoothing term. In general, you should not expect similar results from the two procedures. The following subsections summarize the differences. For more information about the GAM procedure, see Chapter 43, “The GAM Procedure.”

Constructing Spline Basis Expansions

The GAMPL procedure uses thin-plate regression splines to construct basis expansions for each spline term, and each term allows multiple variables. The GAM procedure uses univariate or bivariate smoothing splines to construct basis expansions, and each term allows only one or two variables. The thin-plate regression splines that PROC GAMPL uses are low-rank approximations to multivariate smoothing splines. The GAM procedure also allows loess smoothers.

Fitting Generalized Additive Models

The GAMPL procedure fits a generalized additive model that has fixed smoothing parameters by using a global design matrix and a roughness penalty matrix. The GAM procedure uses partial residuals to fit against single smoothing terms. For models that have unknown smoothing parameters, the GAMPL procedure estimates smoothing parameters simultaneously by optimizing global criteria such as generalized cross validation (GCV) and the unbiased risk estimator (UBRE). The GAM procedure estimates each smoothing parameter by optimizing the local criterion GCV one spline term at a time.
Distribution Families and Link Functions

The GAMPL procedure supports all the distribution families and all the link functions that the GAM procedure supports. In addition, PROC GAMPL fits models in the negative binomial and Tweedie families. PROC GAMPL supports any link function for each distribution, whereas PROC GAM supports only the canonical link for each distribution.

Testing Smoothing Components

The GAMPL procedure tests the total contribution for a spline term, including both linear and nonlinear trends. The GAM procedure tests the existence of nonlinearity for a spline term beyond the linear trend.

Model Inference

A global Bayesian posterior covariance matrix is available for models that are fitted by the GAMPL procedure. The confidence limits for prediction of each observation are available, in addition to component-wise confidence limits. For generalized additive models that are fitted by the GAM procedure, only the component-wise confidence limits are available, and they are based on the partial residuals for each smoothing term. The degrees of freedom for generalized additive models that are fitted by the GAMPL procedure is defined as the trace of the degrees-of-freedom matrix. The degrees of freedom for generalized additive models that are fitted by the GAM procedure is approximated by summing the trace of the smoothing matrix for each smoothing term.

Getting Started: GAMPL Procedure

This example concerns the proportions and demographic and geographic characteristics of votes that were cast in 3,107 counties in the United States in the 1980 presidential election. You can use the data set sashelp.Vote1980 directly from the SASHELP library or download it from the StatLib Datasets Archive (Vlachos 1998). For more information about the data set, see Pace and Barry (1997).

The data set contains 3,107 observations and seven variables. The dependent variable LogVoteRate is the logarithm transformation of the proportion of the county population who voted for any candidate. The six explanatory variables are the number of people in the county 18 years of age or older (Pop), the number of people in the county who have a 12th-grade or higher education (Edu), the number of owned housing units (Houses), the aggregate income (Income), and the scaled longitude and latitude of geographic centroids (Longitude and Latitude).

The following statements produce the plot of LogVoteRate with respect to the geographic locations Longitude and Latitude:

```sas
%let off0 = offsetmin=0 offsetmax=0
linearopts=(thresholdmin=0 thresholdmax=0);
proc template;
    define statgraph surface;
        dynamic _title _z;
        begingraph / designwidth=defaultDesignHeight;
            entrytitle _title;
            layout overlay / xaxisopts=(&off0) yaxisopts=(&off0);
                contourplotparm z=_z y=Latitude x=Longitude / gridded=FALSE;
    endgraph;
end;
```
The objective is to explore the nonlinear dependency structure between the dependent variable and demographic variables (Pop, Edu, Houses, and Income), in addition to the spatial variations on geographic variables (Longitude and Latitude). The following statements use thin-plate regression splines to fit a generalized additive model:
ods graphics on;

proc gampl data=sashelp.Vote1980 plots seed=12345;
   model LogVoteRate = spline(Pop ) spline(Edu) spline(Houses)
                  spline(Income) spline(Longitude Latitude);
   id Longitude Latitude;
   output out=VotePred;
run;

With ODS Graphics enabled by the first statement, the PLOTS option in the PROC GAMPL statement requests a smoothing component panel of fitted spline terms. The SEED option specifies the random seed so that you can reproduce the analysis.

The default output from this analysis is presented in Figure 44.2 through Figure 44.10.

The “Performance Information” table in Figure 44.2 shows that PROC GAMPL executed in single-machine mode (that is, on the server where SAS is installed). When high-performance procedures run in single-machine mode, they use concurrently scheduled threads. In this case, four threads were used.

![Figure 44.2 Performance Information](image)

The GAMPL Procedure

<table>
<thead>
<tr>
<th>Performance Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution Mode</td>
</tr>
<tr>
<td>Number of Threads</td>
</tr>
</tbody>
</table>

Figure 44.3 displays the “Model Information” table. The response variable LogVoteRate is modeled by using a normal distribution whose mean is modeled by an identity link function. The GAMPL procedure uses the performance iteration method and the GCV criterion as the fitting criterion. PROC GAMPL searches for the optimum smoothing parameters by using the Newton-Raphson algorithm to optimize the fitting criterion. The random number seed is set to 12,345. Random number generation is used for sampling from observations to form spline knots and truncated eigendecomposition. Changing the random number seed might yield slightly different model fits.

![Figure 44.3 Model Information](image)

<table>
<thead>
<tr>
<th>Model Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source</td>
</tr>
<tr>
<td>Response Variable</td>
</tr>
<tr>
<td>Distribution</td>
</tr>
<tr>
<td>Link Function</td>
</tr>
<tr>
<td>Fitting Method</td>
</tr>
<tr>
<td>Fitting Criterion</td>
</tr>
<tr>
<td>Optimization Technique for Smoothing</td>
</tr>
<tr>
<td>Random Number Seed</td>
</tr>
</tbody>
</table>

Figure 44.4 displays the “Number of Observations” table. All 3,107 observations in the data set are used in the analysis. For data sets that have missing or invalid values, the number of used observations might be less than the number of observations read.
Figure 44.4 Number of Observations

<table>
<thead>
<tr>
<th>Number of Observations Read</th>
<th>3107</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Observations Used</td>
<td>3107</td>
</tr>
</tbody>
</table>

Figure 44.5 displays the convergence status of the performance iteration method.

Figure 44.5 Convergence Status

The performance iteration converged after 3 steps.

Figure 44.6 shows the “Fit Statistics” table. The penalized log likelihood and the roughness penalty are displayed. You can use effective degrees of freedom to compare generalized additive models with generalized linear models that do not have spline transformations. Information criteria such as Akaike’s information criterion (AIC), Akaike’s bias-corrected information criterion (AICC), and Schwarz Bayesian information criterion (BIC) can also be used for comparisons. These criteria penalize the –2 log likelihood for effective degrees of freedom. The GCV criterion is used to compare against other generalized additive models or models that are penalized.

Figure 44.6 Fit Statistics

<table>
<thead>
<tr>
<th>Fit Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Penalized Log Likelihood</td>
</tr>
<tr>
<td>Roughness Penalty</td>
</tr>
<tr>
<td>Effective Degrees of Freedom</td>
</tr>
<tr>
<td>Effective Degrees of Freedom for Error</td>
</tr>
<tr>
<td>AIC (smaller is better)</td>
</tr>
<tr>
<td>AICC (smaller is better)</td>
</tr>
<tr>
<td>BIC (smaller is better)</td>
</tr>
<tr>
<td>GCV (smaller is better)</td>
</tr>
</tbody>
</table>

The “Parameter Estimates” table in Figure 44.7 shows the regression parameter and dispersion parameter estimates. In this model, the intercept is the only regression parameter because (1) all variables are characterized by spline terms and no parametric effects are present and (2) the intercept absorbs the constant effect that is extracted from each spline term to make fitted splines identifiable. The dispersion parameter is estimated by maximizing the likelihood, given other model parameters.

Figure 44.7 Regression Parameter Estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>DF</th>
<th>Estimate</th>
<th>Standard Error</th>
<th>Chi-Square</th>
<th>Pr &gt; ChiSq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>1</td>
<td>-0.576234</td>
<td>0.001803</td>
<td>102119.645</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Dispersion</td>
<td>1</td>
<td>0.010103</td>
<td>0.014287</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The “Estimates for Smoothing Components” table is shown in Figure 44.8. For each spline term, the effective degrees of freedom, the estimated smoothing parameter, and the corresponding roughness penalty are displayed. The table also displays additional information about spline terms, such as the number of parameters, penalty matrix rank, and number of spline knots.

**Figure 44.8** Estimates for Smoothing Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Effective DF</th>
<th>Smoothing Parameter</th>
<th>Roughness Penalty</th>
<th>Number of Parameters</th>
<th>Rank of Penalty Matrix</th>
<th>Number of Knots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline(Pop)</td>
<td>7.80559</td>
<td>0.0398</td>
<td>0.0114</td>
<td>9</td>
<td>10</td>
<td>2000</td>
</tr>
<tr>
<td>Spline(Edu)</td>
<td>7.12453</td>
<td>0.2729</td>
<td>0.0303</td>
<td>9</td>
<td>10</td>
<td>2000</td>
</tr>
<tr>
<td>Spline(Houses)</td>
<td>7.20940</td>
<td>0.1771</td>
<td>0.0370</td>
<td>9</td>
<td>10</td>
<td>2000</td>
</tr>
<tr>
<td>Spline(Income)</td>
<td>5.92854</td>
<td>0.7498</td>
<td>0.0488</td>
<td>9</td>
<td>10</td>
<td>2000</td>
</tr>
<tr>
<td>Spline(Longitude Latitude)</td>
<td>18.64138</td>
<td>0.000359</td>
<td>0.1304</td>
<td>19</td>
<td>20</td>
<td>2000</td>
</tr>
</tbody>
</table>

Figure 44.9 displays the hypothesis testing results for each smoothing component. The null hypothesis for each spline term is whether the total dependency on each variable is 0. The effective degrees of freedom for both fit and test is displayed.

**Figure 44.9** Tests for Smoothing Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Effective DF</th>
<th>Effective DF for Test</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline(Pop)</td>
<td>7.80559</td>
<td>8</td>
<td>1443.64</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Spline(Edu)</td>
<td>7.12453</td>
<td>8</td>
<td>153.94</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Spline(Houses)</td>
<td>7.20940</td>
<td>8</td>
<td>1213.94</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Spline(Income)</td>
<td>5.92854</td>
<td>7</td>
<td>43.17</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Spline(Longitude Latitude)</td>
<td>18.64138</td>
<td>19</td>
<td>1619.15</td>
<td>&lt;.0001</td>
</tr>
</tbody>
</table>

Figure 44.10 displays the “Smoothing Component Panel” for all the spline terms used in the model. It displays predicted spline curves and 95% Bayesian posterior confidence bands for each univariate spline term.
The preceding program contains an ID statement and an OUTPUT statement. The ID statement transfers the specified variables (Longitude and Latitude) from the input data set, sashelp.Vote1980, to the output data set, VotePred. The OUTPUT statement requests the prediction for each observation by default and saves the results in the data set VotePred. The following run of the SGRENDER procedure produces the fitted surface of the log vote proportion in the 1980 presidential election:

```sas
proc sgrender data=VotePred template=surface;
   dynamic _title='Predicted US County Vote Proportion in the 1980 Election'
      _z = 'Pred';
run;
```

*Figure 44.11* shows the map of predictions of the logarithm transformation of the proportion of county population who voted for any candidates in the 1980 US presidential election from the fitted generalized additive model.
Figure 44.11  Predicted US County Vote Proportion in the 1980 Election

Compared to the map of the logarithm transformations of the proportion of votes cast shown in Figure 44.1, the map of the predictions of the logarithm transformations of the proportion of votes cast has a smoother surface.
Syntax: GAMPL Procedure

The following statements are available in the GAMPL procedure:

```
PROC GAMPL <options>;
   CLASS variable <(options)> . . . <variable <(options)>> </ global-options> ;
   MODEL response <(response-options)> = <PARAM(effects)>
      < spline-effects > </ model-options> ;
   MODEL events / trials = <PARAM(effects)> < spline-effects > </ model-options> ;
   OUTPUT <OUT=SAS-data-set> < keyword < =name>> . . . <keyword < =name>> </ options> ;
   PERFORMANCE performance-options ;
   FREQ variable ;
   ID variables ;
   WEIGHT variable ;
```

The PROC GAMPL statement and at least one MODEL statement are required. The CLASS statement can appear multiple times. If a CLASS statement is specified, it must precede the MODEL statements. The following sections describe the PROC GAMPL statement and then describe the other statements in alphabetical order.

PROC GAMPL Statement

```
PROC GAMPL <options> ;
```

The PROC GAMPL statement invokes the procedure. Table 44.1 summarizes the available options in the PROC GAMPL statement by function. The options are then described fully in alphabetical order.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Options</td>
<td></td>
</tr>
<tr>
<td>ALPHA=</td>
<td>Specifies a global significance level</td>
</tr>
<tr>
<td>DATA=</td>
<td>Specifies the input data set</td>
</tr>
<tr>
<td>NAMELEN=</td>
<td>Limits the length of effect names</td>
</tr>
<tr>
<td>SEED=</td>
<td>Sets the seed for pseudorandom number generation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Display Options</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ITDETAILS</td>
<td>Displays the “Iteration History” table</td>
</tr>
<tr>
<td>NOCLPRINT</td>
<td>Limits or suppresses the display of classification variable levels</td>
</tr>
<tr>
<td>NOPRINT</td>
<td>Suppresses ODS output</td>
</tr>
<tr>
<td>PLOTS=</td>
<td>Controls plots that are produced through ODS Graphics</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optimization Subject Options</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PLIKEOPTIONS</td>
<td>Sets optimization parameters for likelihood estimation</td>
</tr>
<tr>
<td>SMOOTHOPTIONS</td>
<td>Sets optimization parameters for smoothing parameter estimation</td>
</tr>
</tbody>
</table>
Table 44.1 continued

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tolerance Options</strong></td>
<td></td>
</tr>
<tr>
<td>SINGCHOL=</td>
<td>Tunes the singularity criterion for Cholesky decompositions</td>
</tr>
<tr>
<td>SINGSWEEP=</td>
<td>Tunes the singularity criterion for the sweep operator</td>
</tr>
<tr>
<td>SINGULAR=</td>
<td>Tunes the general singularity criterion</td>
</tr>
<tr>
<td><strong>User-Defined Format Options</strong></td>
<td></td>
</tr>
<tr>
<td>FMTLIBXML=</td>
<td>Specifies the file reference for a format stream</td>
</tr>
</tbody>
</table>

You can specify the following options in the PROC GAMPL statement.

**ALPHA=** number

specifies a global significance level for the hypothesis testing of smoothing components and the construction of Bayesian confidence bands of predictions. The confidence level is $1 - \text{number}$. The value of \text{number} must be between 0 and 1; the default is 0.05. You can override this global significance level for Bayesian confidence bands of predictions by specifying the \text{ALPHA=} option in the \text{OUTPUT} statement.

**DATA=** SAS-data-set

names the input SAS data set for PROC GAMPL to use. The default is the most recently created data set.

If the procedure executes in distributed mode, the input data are distributed to memory on the appliance nodes and analyzed in parallel, unless the data are already distributed in the appliance database. In that case, PROC GAMPL reads the data alongside the distributed database. For information about the various execution modes, see the section “Processing Modes” (Chapter 2, *SAS/STAT User’s Guide: High-Performance Procedures*). For information about the alongside-the-database model, see the section “Alongside-the-Database Execution” (Chapter 2, *SAS/STAT User’s Guide: High-Performance Procedures*).

**FMTLIBXML=** file-ref

specifies the file reference for the XML stream that contains user-defined format definitions. User-defined formats are handled differently in a distributed computing environment than they are in other SAS products. For information about how to generate an XML stream for your formats, see the section “Working with Formats” (Chapter 2, *SAS/STAT User’s Guide: High-Performance Procedures*).

**ITDETAILS**

adds to the “Iteration History” table the current values of the parameter estimates and their gradients. If the optimization algorithm is used to determine at least one smoothing parameter, the table lists values for smoothing parameters. If all smoothing parameters are fixed or a parametric generalized linear model is specified, the table lists values for regression parameters. These quantities are reported only for parameters that participate in the optimization.

**NAMELEN=** number

specifies the length to which long effect names are shortened. The value of \text{number} must be an integer greater than or equal to 20. By default, \text{NAMELEN}=20.
NOCLPRINT\(<=number>\)
suppresses the display of the “Class Level Information” table if you do not specify \textit{number}. If you specify \textit{number}, the values of the classification variables are displayed for only those variables whose number of levels is less than \textit{number}. Specifying a \textit{number} helps reduce the size of the “Class Level Information” table if some classification variables have a large number of levels.

NOPRINT
suppresses the generation of ODS output.

PLIKEOPTIONS\((\text{optimization-parameters})\)
sets optimization parameters for either maximum or penalized likelihood estimation. For more information about how to specify \textit{optimization-parameters}, see the section “Optimization Parameters” on page 3042.

PLOTS \(<(\text{global-plot-option})><=\text{plot-requests}<\text{(option)}>\)
controls the plots that are produced through ODS Graphics. When ODS Graphics is enabled, PROC GAMPL produces by default a panel of plots of partial prediction curves or surfaces of smoothing components.

ODS Graphics must be enabled before plots can be requested. For example:

\begin{verbatim}
ods graphics on;
proc gampl plots;
    model y=spline(x1) spline(x2);
run;
ods graphics off;
\end{verbatim}

For more information about enabling and disabling ODS Graphics, see the section “Enabling and Disabling ODS Graphics” on page 615 in Chapter 21, “Statistical Graphics Using ODS.”

You can specify the following \textit{global-plot-option}, which applies to the smoothing component plots that the GAMPL procedure generates:

UNPACK | UNPACKPANEL
suppresses paneling. By default, multiple smoothing component plots can appear in some output panels. Specify UNPACK to get each plot individually.

You can specify the following \textit{plot-requests}:

ALL
requests that all default plots be produced.

COMPONENTS \(<\text{(component-option)}>\)
plots a panel of smoothing components of the fitted model. You can specify the following \textit{component-option}:

COMMONAXES
requests that the smoothing component plots use a common vertical axis except for bivariate contour plots. This option enables you to visually judge the relative effect size.
NONE suppresses all plots.

SEED=number

specifies an integer that is used to start the pseudorandom number generator for subset sampling from observations to form knots if necessary and for truncated eigendecomposition. If you do not specify this option, or if number ≤ 0, the seed is generated from the time of day, which is read from the computer’s clock.

SINGCHOL=number

tunes the singularity criterion in Cholesky decompositions. The default is 1E4 times the machine epsilon; this product is approximately 1E–12 on most computers.

SINGSWEEP=number

tunes the singularity criterion in sweep operations that determine collinearity between spline basis expansions. The default is 1E–8.

SINGULAR=number

tunes the general singularity criterion that is applied in sweep and inversion operations. The default is 1E4 times the machine epsilon; this product is approximately 1E–12 on most computers.

SMOOTHOPTIONS(optimization-parameters)

specifies optimization parameters for smoothing parameter estimation. For more information about how to specify optimization-parameters, see the section “Optimization Parameters” on page 3042.

Optimization Parameters

You can specify optimization-parameters for both the PLIKEOPTIONS and SMOOTHOPTIONS options. Depending on the modeling context, some optimization parameters might have no effect. For parametric generalized linear models or generalized additive models that have fixed smoothing parameters, any optimization parameters that you specify in the SMOOTHOPTIONS option are ignored. For the performance iteration method, only the ABSFCONV=, FCONV=, and MAXITER= options are effective for PLIKEOPTIONS. The optimization algorithm is considered to have converged when any one of the convergence criteria that are specified in optimization-parameters is satisfied. Table 44.2 lists the available optimization parameters for both the PLIKEOPTIONS and SMOOTHOPTIONS options.

Table 44.2 Optimization Parameters

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSCONV=</td>
<td>Tunes the absolute function convergence criterion</td>
</tr>
<tr>
<td>ABSFCNV=</td>
<td>Tunes the absolute function difference convergence criterion</td>
</tr>
<tr>
<td>ABSGCONV=</td>
<td>Tunes the absolute gradient convergence criterion</td>
</tr>
<tr>
<td>FCONV=</td>
<td>Tunes the relative function difference convergence criterion</td>
</tr>
<tr>
<td>GCONV=</td>
<td>Tunes the relative gradient convergence criterion</td>
</tr>
<tr>
<td>MAXFUNC=</td>
<td>Specifies the maximum number of function evaluations in any optimization</td>
</tr>
<tr>
<td>MAXITER=</td>
<td>Chooses the maximum number of iterations in any optimization</td>
</tr>
<tr>
<td>MAXTIME=</td>
<td>Specifies the upper limit of CPU time (in seconds) for any optimization</td>
</tr>
<tr>
<td>MINITER=</td>
<td>Specifies the minimum number of iterations in any optimization</td>
</tr>
<tr>
<td>TECHNIQUE=</td>
<td>Selects the optimization technique</td>
</tr>
</tbody>
</table>
You can specify the following optimization-parameters:

**ABSCONV=**\(r\)

**ABSTOL=**\(r\)

Specifies an absolute function convergence criterion. For minimization, termination requires \(f(\psi^{(k)}) \leq r\), where \(\psi\) is the vector of parameters in the optimization and \(f(\cdot)\) is the objective function. The default value of \(r\) is the negative square root of the largest double-precision value, which serves only as a protection against overflow.

**ABSFCONV=**\(r < n\)

**ABSFTOL=**\(r < n\)

Specifies an absolute function difference convergence criterion. For all techniques except NMSIMP, termination requires a small change of the function value in successive iterations,

\[
|f(\psi^{(k-1)}) - f(\psi^{(k)})| \leq r
\]

where \(\psi\) denotes the vector of parameters that participate in the optimization and \(f(\cdot)\) is the objective function. The same formula is used for the NMSIMP technique, but \(\psi^{(k)}\) is defined as the vertex that has the lowest function value and \(\psi^{(k-1)}\) is defined as the vertex that has the highest function value in the simplex. The optional integer value \(n\) specifies the number of successive iterations for which the criterion must be satisfied before the process terminates. By default, **ABSFCONV**=0.

**ABSGCONV=**\(r < n\)

**ABSGTOL=**\(r < n\)

Specifies an absolute gradient convergence criterion. Termination requires the maximum absolute gradient element to be small,

\[
\max_j |g_j(\psi^{(k)})| \leq r
\]

where \(\psi\) denotes the vector of parameters that participate in the optimization and \(g_j(\cdot)\) is the gradient of the objective function with respect to the \(j\)th parameter. This criterion is not used by the NMSIMP technique. The default value is \(r = 1E^{-8}\). The optional integer value \(n\) specifies the number of successive iterations for which the criterion must be satisfied before the process terminates.

**FCONV=**\(r < n\)

**FTOL=**\(r < n\)

Specifies a relative function difference convergence criterion. For all techniques except NMSIMP, termination requires a small relative change of the function value in successive iterations,

\[
\frac{|f(\psi^{(k)}) - f(\psi^{(k-1)})|}{|f(\psi^{(k-1)})|} \leq r
\]

where \(\psi\) denotes the vector of parameters that participate in the optimization and \(f(\cdot)\) is the objective function. The same formula is used for the NMSIMP technique, but \(\psi^{(k)}\) is defined as the vertex that has the lowest function value and \(\psi^{(k-1)}\) is defined as the vertex that has the highest function value in the simplex.

The default value is \(r = 2 \times \epsilon\), where \(\epsilon\) is the machine precision. The optional integer value \(n\) specifies the number of successive iterations for which the criterion must be satisfied before the process terminates.
GCONV=r < n>

GTOL=r < n>

specifies a relative gradient convergence criterion. For all techniques except CONGRA and NMSIMP, termination requires that the normalized predicted function reduction be small,

\[
\frac{g(\psi^{(k)})[H^{(k)}]^{-1}g(\psi^{(k)})}{|f(\psi^{(k)})|} \leq r
\]

where \( \psi \) denotes the vector of parameters that participate in the optimization, \( f(\cdot) \) is the objective function, and \( g(\cdot) \) is the gradient. For the CONGRA technique (where a reliable Hessian estimate, \( H \), is not available), the following criterion is used:

\[
\frac{\| g(\psi^{(k)}) \|_2^2 \| s(\psi^{(k)}) \|_2}{\| g(\psi^{(k)}) - g(\psi^{(k-1)}) \|_2 \| f(\psi^{(k)}) \|} \leq r
\]

This criterion is not used by the NMSIMP technique. The default value is \( r = 1E^{-8} \). The optional integer value \( n \) specifies the number of successive iterations for which the criterion must be satisfied before the process terminates.

MAXFUNC=n

MAXFU=n

specifies the maximum number of function calls in the optimization process. The default values are as follows, depending on the optimization technique that you specify in the TECHNIQUE= option:

- TRUREG, NRRIDG, NEWRAP: \( n = 125 \)
- QUANEW, DBLDOG: \( n = 500 \)
- CONGRA: \( n = 1,000 \)
- NMSIMP: \( n = 3,000 \)

The optimization can terminate only after completing a full iteration. Therefore, the number of function calls that are actually performed can exceed \( n \).

MAXITER=n

MAXIT=n

specifies the maximum number of iterations in the optimization process. The default values are as follows, depending on the optimization technique that you specify in the TECHNIQUE option:

- TRUREG, NRRIDG, NEWRAP: \( n = 50 \)
- QUANEW, DBLDOG: \( n = 200 \)
- CONGRA: \( n = 400 \)
- NMSIMP: \( n = 1,000 \)

These default values also apply when \( n \) is specified as a missing value.
MAXTIME=r
specifies an upper limit of $r$ seconds of CPU time for the optimization process. The default value is the largest floating-point double representation of your computer. The time that you specify in this option is checked only once at the end of each iteration. Therefore, the actual running time can be longer than $r$.

MINITER=n
MINIT=n
specifies the minimum number of iterations. If you request more iterations than are actually needed for convergence to a stationary point, the optimization algorithms might behave strangely. For example, the effect of rounding errors can prevent the algorithm from continuing for the required number of iterations. By default, MINITER=0.

TECHNIQUE=keyword
TECH=keyword
specifies the optimization technique for obtaining smoothing parameter estimates and regression parameter estimates. You can choose from the following techniques:

- CONGRA performs a conjugate-gradient optimization.
- DBLDOG performs a version of double-dogleg optimization.
- NEWRAP performs a Newton-Raphson optimization with line search.
- NMSIMP performs a Nelder-Mead simplex optimization.
- NONE performs no optimization.
- NRRIDG performs a Newton-Raphson optimization with ridging.
- QUANEW performs a dual quasi-Newton optimization.
- TRUREG performs a trust-region optimization

By default, TECHNIQUE=NEWRAP for the performance iteration (METHOD=PERFORMANCE), and TECHNIQUE=QUANEW for the outer iteration (METHOD=OUTER).

For more information, see the section “Choosing an Optimization Technique” on page 3068.

---

**CLASS Statement**

CLASS variable <(options)> . . . < variable <(options)> > < / global-options > ;

The CLASS statement names the classification variables to be used as explanatory variables in the analysis. The CLASS statement must precede the MODEL statement. You can list the response variable for binary models in the CLASS statement, but this is not necessary.

FREQ Statement

FREQ variable ;

The variable in the FREQ statement identifies a numeric variable in the data set that contains the frequency of occurrence of each observation. PROC GAMPL treats each observation as if it appeared \( f \) times, where the frequency value \( f \) is the value of the FREQ variable for the observation. If \( f \) is not an integer, then \( f \) is truncated to an integer. If \( f \) is less than 1 or missing, the observation is not used in the analysis. When you do not specify the FREQ statement, each observation is assigned a frequency of 1.

ID Statement

ID variables ;

The ID statement lists one or more variables from the input data set that are to be transferred to the output data set that you specify in the OUTPUT statement.

For more information, see the section “ID Statement” (Chapter 3, SAS/STAT User’s Guide: High-Performance Procedures).

MODEL Statement

MODEL response < (response-options) > = < PARAM(effects) > < spline-effects > </ model-options > ;
MODEL events / trials = < PARAM(effects) > < spline-effects > </ model-options > ;

The MODEL statement specifies the response (dependent or target) variable and the predictor (independent or explanatory) effects of the model. You can specify the response in the form of a single variable or in the form of a ratio of two variables, which are denoted \( \text{events/trials} \). The first form applies to all distribution families; the second form applies only to summarized binomial response data. When you have binomial data, the \( \text{events} \) variable contains the number of positive responses (or events) and the \( \text{trials} \) variable contains the number of trials. The values of both \( \text{events} \) and \( \text{trials} - \text{events} \) must be nonnegative, and the value of \( \text{trials} \) must be positive. If you specify a single \( \text{response} \) variable that is in a CLASS statement, then the response is assumed to be binary.

You can specify parametric effects that are constructed from variables in the input data set and include the effects in the parentheses of a PARAM( ) option, which can appear multiple times. For information about constructing the model effects, see the section “Specification and Parameterization of Model Effects” (Chapter 3, SAS/STAT User’s Guide: High-Performance Procedures).

You can specify spline-effects by including independent variables inside the parentheses of the SPLINE( ) option. Only continuous variables (not classification variables) can be specified in spline-effects. Each spline-effect can have at least one variable and optionally some spline-options. You can specify any number of spline-effects. The following table shows some examples.
MODEL Statement

Spline Effect Specification Meaning

*Spline(x)* Constructs the univariate spline with x and uses the observed data points as knots. The maximum degrees of freedom is 10. PROC GAMPL uses an optimization algorithm to determine the optimal smoothing parameter.

*Spline(x1/knots=list(1 to 10))* Constructs the univariate spline by using x1 and a supplied list of knots from 1 to 10. PROC GAMPL uses an optimization algorithm to determine the optimal smoothing parameter.

*Spline(x2 x3/smooth=0.3)* Constructs the bivariate spline by using x2 and x3 and a fixed smoothing parameter 0.3.

*Spline(x4 x5 x6/maxdf=40)* Constructs the trivariate spline by using x4, x5, and x6 and a maximum of 40 degrees of freedom. PROC GAMPL uses an optimization algorithm to determine the optimal smoothing parameter.

Both parametric effects and spline effects are optional. If none are specified, a model that contains only an intercept is fitted. If only parametric effects are present, PROC GAMPL fits a parametric generalized linear model by using the terms inside the parentheses of all PARAM( ) terms. If only spline effects are present, PROC GAMPL fits a nonparametric additive model. If both types of effects are present, PROC GAMPL fits a semiparametric model by using the parametric effects as the linear part of the model.

There are three sets of options in the MODEL statement. The **response-options** determine how the GAMPL procedure models probabilities for binary data. The **spline-options** controls how each spline term forms basis expansions. The **model-options** control other aspects of model formation and inference. Table 44.3 summarizes these options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Response Variable Options for Binary Models</strong></td>
<td></td>
</tr>
<tr>
<td>DESCENDING</td>
<td>Reverses the response categories</td>
</tr>
<tr>
<td>EVENT=</td>
<td>Specifies the event category</td>
</tr>
<tr>
<td>ORDER=</td>
<td>Specifies the sort order</td>
</tr>
<tr>
<td>REF=</td>
<td>Specifies the reference category</td>
</tr>
<tr>
<td><strong>Smoothing Options for Spline Effects</strong></td>
<td></td>
</tr>
<tr>
<td>DETAILS</td>
<td>Requests detailed spline information</td>
</tr>
<tr>
<td>DF=</td>
<td>Specifies the fixed degrees of freedom</td>
</tr>
<tr>
<td>INITSMOOTH=</td>
<td>Specifies the starting value for the smoothing parameter</td>
</tr>
<tr>
<td>KNOTS=</td>
<td>Specifies the knots to be used for constructing the spline</td>
</tr>
<tr>
<td>M=</td>
<td>Specifies polynomial orders for constructing the spline</td>
</tr>
<tr>
<td>MAXDF=</td>
<td>Specifies the maximum degrees of freedom</td>
</tr>
<tr>
<td>MAXKNOTS=</td>
<td>Specifies the maximum number of knots to be used for constructing the spline</td>
</tr>
<tr>
<td>MAXSMOOTH=</td>
<td>Specifies the upper bound for the smoothing parameter</td>
</tr>
</tbody>
</table>
Table 44.3 continued

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINSMOOTH=</td>
<td>Specifies the lower bound for the smoothing parameter</td>
</tr>
<tr>
<td>SMOOTH=</td>
<td>Specifies a fixed smoothing parameter</td>
</tr>
</tbody>
</table>

Model Options

- **ALLOBS** Requests all nonmissing values of spline variables for constructing spline basis functions regardless of other model variables.
- **CRITERION=** Specifies the model evaluation criterion.
- **DISPERSION | PHI=** Specifies the fixed dispersion parameter.
- **DISTRIBUTION | DIST=** Specifies the response distribution.
- **FDHESSIAN** Requests a finite-difference Hessian for smoothing parameter selection.
- **INITIALPHI=** Specifies the starting value of the dispersion parameter.
- **LINK=** Specifies the link function.
- **NORMALIZE** Requests normalized spline basis functions for model fitting.
- **MAXPHI=** Specifies the upper bound for searching the dispersion parameter.
- **METHOD=** Specifies the algorithm for selecting smoothing parameters.
- **MINPHI=** Specifies the lower bound for searching the dispersion parameter.
- **OFFSET=** Specifies the offset variable.
- **RIDGE=** Specifies the ridge parameter.
- **SCALE=** Specifies the method for estimating the dispersion parameter.

Response Variable Options

Response variable options determine how the GAMPL procedure models probabilities for binary data.

You can specify the following *response-options* by enclosing them in parentheses after the *response* variable.

**DESCENDING**

**DESC** reverses the order of the response categories. If you specify both the DESCENDING and **ORDER=** options, PROC GAMPL orders the response categories according to the **ORDER=** option and then reverses that order.

**EVENT='category' | FIRST | LAST** specifies the event category for the binary response model. PROC GAMPL models the probability of the event category. The EVENT= option has no effect when there are more than two response categories.

You can specify any of the following:

**'category'** specifies that observations whose value matches *category* (formatted, if a format is applied) in quotation marks represent events in the data. For example, the following statements specify that observations that have a formatted value of ‘1’ represent events in the data. The probability that is modeled by the GAMPL procedure is thus the probability that the variable *def* takes on the (formatted) value ‘1’.
**MODEL Statement**

```sas
proc gampl data=MyData;
  class A B C;
  model def(event = '1') = param(A B C) spline(x1 x2 x3);
run;
```

**FIRST**

designates the first ordered category as the event.

**LAST**

designates the last ordered category as the event.

By default, EVENT=FIRST.

**ORDER=** DATA | FORMATTED | INTERNAL

**ORDER=FREQ | FREQDATA | FREQFORMATTED | FREQINTERNAL**

specifies the sort order for the levels of the response variable. When ORDER=FORMATTED (the default) for numeric variables for which you have supplied no explicit format (that is, for which there is no corresponding FORMAT statement in the current PROC GAMPL run or in the DATA step that created the data set), the levels are ordered by their internal (numeric) value. Table 44.4 shows the interpretation of the ORDER= option.

<table>
<thead>
<tr>
<th>ORDER=</th>
<th>Levels Sorted By</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>Order of appearance in the input data set</td>
</tr>
<tr>
<td>FORMATTED</td>
<td>External formatted value, except for numeric variables that have no explicit format, which are sorted by their unformatted (internal) value</td>
</tr>
<tr>
<td>FREQ</td>
<td>Descending frequency count (levels that have the most observations come first in the order)</td>
</tr>
<tr>
<td>FREQDATA</td>
<td>Order of descending frequency count; within counts by order of appearance in the input data set when counts are tied</td>
</tr>
<tr>
<td>FREQFORMATTED</td>
<td>Order of descending frequency count; within counts by formatted value when counts are tied</td>
</tr>
<tr>
<td>FREQINTERNAL</td>
<td>Order of descending frequency count; within counts by unformatted value when counts are tied</td>
</tr>
<tr>
<td>INTERNAL</td>
<td>Unformatted value</td>
</tr>
</tbody>
</table>

By default, ORDER=FORMATTED. For the FORMATTED and INTERNAL orders, the sort order is machine-dependent.

For more information about sort order, see the chapter about the SORT procedure in *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*. 
**REF='category' | FIRST | LAST**

specifies the reference category for the binary response model. Specifying one response category as the reference is the same as specifying the other response category as the event category. You can specify any of the following:

- **'category'**
  - specifies that observations whose value matches `category` (formatted, if a format is applied) are designated as the reference.
- **FIRST**
  - designates the first ordered category as the reference
- **LAST**
  - designates the last ordered category as the reference.

By default, REF=LAST.

**Spline Options**

- **DETAILS**
  - requests a detailed spline specification information table.
- **DF=number**
  - specifies a fixed degrees of freedom. When you specify this option, no smoothing parameter selection is performed on the spline term. If `number` is not an integer, then `number` is truncated to an integer.
- **INITSMOOTH=number**
  - specifies the starting value for a smoothing parameter. The `number` must be nonnegative.
- **KNOTS=method**
  - specifies the method for supplying user-defined knot values instead of using data values for constructing basis expansions. You can use the following methods for supplying the knots:
  - **LIST(list-of-values)**
    - specifies a list of values as knots for the spline construction. For a multivariate spline term, the listed values are taken as multiple row vectors, where each vector has values that are ordered by specified variables. If the last row vector of knots contains fewer values than the number of variables, then the last row vector is ignored. For example, the following specification of a spline term produces two actual knot vectors ($k_1$ and $k_2$) and the value 5 is ignored.

  ```
spline(x1 x2/knots=list(1 2 3 4 5))
  ```

<table>
<thead>
<tr>
<th>Table 44.5</th>
<th>Knot Values for a Bivariate Spline with a Supplied List</th>
</tr>
</thead>
</table>
  |    x1    |    x2    |-
  | $k_1$   | 1        | 2        |
  | $k_2$   | 3        | 4        |
**EQUAL(n)**
specifies the number of equally spaced interior knots for every variable in a spline term. Two boundary knots are automatically added to the knot list for each variable such that the total number of knots is \((n + 2)^d\), where \(d\) is the number of variables in the spline term. For a multivariate spline term, knot values for each variable are determined independently from the corresponding boundary values. For example, if the boundary points for \(x1\) are 1 and 5 and the boundary points for \(x2\) are 2 and 6, then the following specification of a spline term produces nine actual knots \((k_1 \text{—} k_9)\), which consist of two boundary knots and one interior knot for each variable.

```plaintext
spline(x1 x2/knots=equal(1))
```

<table>
<thead>
<tr>
<th>Table 44.6</th>
<th>Knot Values for a Bivariate Spline with One Interior Knot</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1 x2</td>
<td>x1  x2</td>
</tr>
<tr>
<td>k1</td>
<td>1   2</td>
</tr>
<tr>
<td>k2</td>
<td>1   4</td>
</tr>
<tr>
<td>k3</td>
<td>1   6</td>
</tr>
<tr>
<td>k4</td>
<td>3   2</td>
</tr>
<tr>
<td>k5</td>
<td>3   4</td>
</tr>
<tr>
<td>k6</td>
<td>3   6</td>
</tr>
<tr>
<td>k7</td>
<td>5   2</td>
</tr>
<tr>
<td>k8</td>
<td>5   4</td>
</tr>
<tr>
<td>k9</td>
<td>5   6</td>
</tr>
</tbody>
</table>

**M=number**
specifies the order of the derivative in the penalty term. The *number* must be a positive integer. The default is \(\max(2, \int(d/2) + 1)\), where \(d\) is the number of variables in the spline term.

**MAXDF=number**
specifies the maximum *number* of degrees of freedom. When a thin-plate regression spline is formed, the specified *number* is used for constructing a low-rank penalty matrix to approximate the penalty matrix via the truncated eigendecomposition. The *number* must be greater than \(\frac{(m^d - 1)}{d}\), where \(m\) is the derivative order that is specified in the M= option. The default is \(10 \times d\). For more information, see the section “Thin-Plate Regression Splines” on page 3057.

**MAXKNOTS=number**
specifies the maximum *number* of knots if data points are used to form knots. If KNOTS=LIST(list-of-values) is not specified, PROC GAMPL forms knots from unique data points. If the number of unique data points is greater than *number*, a subset of size *number* is formed by random sampling from all unique data points. The *number* cannot exceed the largest integer that can be stored on the given machine. By default, MAXKNOTS=2000.

**MAXSMOOTH=number**
specifies the upper bound for the smoothing parameter. The default is the largest double-precision value.
MINSMOOTH=number
  specifies the lower bound for the smoothing parameter. By default, MINSMOOTH=0.

SMOOTH=number
  specifies a fixed smoothing parameter. When you specify this option, no smoothing parameter selection is performed on the spline term.

Model Options

ALLOBS
  requests that all nonmissing values of the variables be specified in a spline term for constructing the spline basis functions, regardless of whether other model variables are missing.

CRITERION=criterion
  specifies the model evaluation criterion for selecting smoothing parameters for spline-effects. You can specify the following values:

GACV<(FACTOR=number | GAMMA=number)>
  uses the generalized approximate cross validation (GACV) criterion to evaluate models.

GCV<(FACTOR=number | GAMMA=number)>
  uses the generalized cross validation (GCV) criterion to evaluate models.

UBRE<(FACTOR=number | GAMMA=number)>
  uses the unbiased risk estimator (UBRE) criterion to evaluate models.

The default criterion depends on the value of the DISTRIBUTION= option. For distributions that involve dispersion parameters, GCV is the default. For distributions without dispersion parameters, UBRE is the default. For all three criteria, you can optionally use the FACTOR= option to specify an extra tuning parameter in order to penalize more for model roughness. The value of number must be greater than or equal to 1. For more information about the model evaluation criteria, see “Model Evaluation Criteria” on page 3062.

DISPERSION=number

PHI=number
  specifies a fixed dispersion parameter for distributions that have a dispersion parameter. The dispersion parameter that is used in all computations is fixed at number; it is not estimated.

DISTRIBUTION=keyword
  specifies the response distribution for the model. The keywords and their associated distributions are shown in Table 44.7.
Table 44.7 Built-In Distribution Functions

<table>
<thead>
<tr>
<th>DISTRIBUTION=</th>
<th>Distribution Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>BINARY</td>
<td>Binary</td>
</tr>
<tr>
<td>BINOMIAL</td>
<td>Binary or binomial</td>
</tr>
<tr>
<td>GAMMA</td>
<td>Gamma</td>
</tr>
<tr>
<td>INVERSEGAUSSIAN</td>
<td>IG</td>
</tr>
<tr>
<td>NEGATIVEBINOMIAL</td>
<td>NB</td>
</tr>
<tr>
<td>NORMAL</td>
<td>GAUSSIAN</td>
</tr>
<tr>
<td>POISSON</td>
<td>Poisson</td>
</tr>
<tr>
<td>TWEEDIE&lt; (Tweedie-options) &gt;</td>
<td>Tweedie</td>
</tr>
</tbody>
</table>

When DISTRIBUTION=TWEEDIE, you can specify the following Tweedie-options:

INITIALP=
  specifies a starting value for iterative estimation of the Tweedie power parameter.

P=
  requests a fixed Tweedie power parameter.

If you do not specify a link function in the LINK= option, a default link function is used. The default link function for each distribution is shown in Table 44.8. You can use any link function shown in Table 44.9 by specifying the LINK= option. Other commonly used link functions for each distribution are shown in Table 44.8.

Table 44.8 Default and Commonly Used Link Functions

<table>
<thead>
<tr>
<th>DISTRIBUTION=</th>
<th>Default Link Function</th>
<th>Other Commonly Used Link Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>BINARY</td>
<td>Logit</td>
<td>Probit, complementary log-log, log-log</td>
</tr>
<tr>
<td>BINOMIAL</td>
<td>Logit</td>
<td>Probit, complementary log-log, log-log</td>
</tr>
<tr>
<td>GAMMA</td>
<td>Reciprocal</td>
<td>Log</td>
</tr>
<tr>
<td>INVERSEGAUSSIAN</td>
<td>IG</td>
<td>Reciprocal square</td>
</tr>
<tr>
<td>NEGATIVEBINOMIAL</td>
<td>NB</td>
<td>Log</td>
</tr>
<tr>
<td>NORMAL</td>
<td>GAUSSIAN</td>
<td>Identity</td>
</tr>
<tr>
<td>POISSON</td>
<td>Log</td>
<td>Log</td>
</tr>
<tr>
<td>TWEEDIE</td>
<td>Log</td>
<td>Log</td>
</tr>
</tbody>
</table>

FDHESSIAN
  requests that the second-order derivatives (Hessian) be computed using finite-difference approximations based on evaluation of the first-order derivatives (gradient). This option might be useful if the analytical Hessian takes a long time to compute.
INITIALPHI=number
specifies a starting value for iterative maximum likelihood estimation of the dispersion parameter for
distributions that have a dispersion parameter.

LINK=keyword
specifies the link function for the model. The keywords and the associated link functions are shown
in Table 44.9. Default and commonly used link functions for the available distributions are shown in
Table 44.8.

<table>
<thead>
<tr>
<th>LINK=</th>
<th>Function</th>
<th>g(μ) = η =</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLOGLOG</td>
<td>Complementary log-log</td>
<td>log(−log(1−μ))</td>
</tr>
<tr>
<td>IDENTITY</td>
<td>Identity</td>
<td>μ</td>
</tr>
<tr>
<td>INV</td>
<td>Reciprocal</td>
<td>1/μ</td>
</tr>
<tr>
<td>INV2</td>
<td>Reciprocal square</td>
<td>1/μ^2</td>
</tr>
<tr>
<td>LOG</td>
<td>Logarithm</td>
<td>log(μ)</td>
</tr>
<tr>
<td>LOGIT</td>
<td>Logit</td>
<td>log(μ/(1−μ))</td>
</tr>
<tr>
<td>LOGLOG</td>
<td>Log-log</td>
<td>−log(−log(μ))</td>
</tr>
<tr>
<td>PROBIT</td>
<td>Probit</td>
<td>Φ^−1(μ)</td>
</tr>
</tbody>
</table>

\(Φ^−1(\cdot)\) denotes the quantile function of the standard normal distribution.

MAXPHI=number
specifies an upper bound for maximum likelihood estimation of the dispersion parameter for distribu-
tions that have a dispersion parameter.

METHOD=OUTER | PERFORMANCE
specifies the algorithm for selecting smoothing parameters for spline-effects. You can specify the
following values:

OUTER specifies the outer iteration method for selecting smoothing parameters. For more
information about the method, see the section “Outer Iteration” on page 3063.

PERFORMANCE specifies the performance iteration method for selecting smoothing parameters.
For more information about the method, see the section “Performance Iteration” on
page 3064.

By default, METHOD=PERFORMANCE.

MINPHI=number
specifies a lower bound for maximum likelihood estimation of the dispersion parameter for distribu-
tions that have a dispersion parameter.

NORMALIZE
requests normalized spline basis functions for model fitting. After the regression spline basis functions
are computed, each column is standardized to have a unit standard error. The corresponding penalty
matrix is also scaled accordingly. This option might be helpful when you have badly scaled data.
OFFSET=variable
specifies a variable to be used as an offset to the linear predictor. An offset plays the role of an effect whose coefficient is known to be 1. The offset variable cannot appear in the CLASS statement or elsewhere in the MODEL statement. Observations that have missing values for the offset variable are excluded from the analysis.

RIDGE=number
allows a ridge parameter such that a diagonal matrix $H: H_{ii} = number$ is added to the optimization problem with respect to regression parameters:

$$\min (y - X\beta)'(y - X\beta) + \beta' S\beta + \beta' HH\beta$$

with respect to $\beta$

By default, RIDGE=0. Specifying a small ridge parameter might be helpful if the model matrix $X'X + S\lambda$ is close to singular.

SCALE=DEVIANCE | MLE | PEARSON
specifies the method for estimating the scale and dispersion parameters. You can specify the following values:

DEVIANCE
estimates the dispersion parameter by using the deviance statistic.

MLE
computes the dispersion parameter by maximizing the likelihood or penalized likelihood.

PEARSON
estimates the dispersion parameter by using Pearson’s statistic.

By default, SCALE=MLE.

OUTPUT Statement

The OUTPUT statement creates a data set that contains observationwise statistics that are computed after the model is fitted. The variables in the input data set are not included in the output data set in order to avoid data duplication for large data sets; however, variables that are specified in the ID statement are included.

If the input data set is distributed (so that accessing data in a particular order cannot be guaranteed), the GAMPL procedure copies the distribution or partition key to the output data set so that its contents can be joined with the input data.

The computation of the output statistics is based on the final parameter estimates. If the model fit does not converge, missing values are produced for the quantities that depend on the estimates.

You can specify the following syntax elements in the OUTPUT statement before the slash (/).
OUT=SAS-data-set
DATA=SAS-data-set

specifies the name of the output data set. If you omit this option, PROC GAMPL uses the DATAn
convention to name the output data set.

keyword <= name >

specifies a statistic to include in the output data set and optionally assigns a name to the variable. If
you do not provide a name, the GAMPL procedure assigns a default name based on the keyword.

You can specify the following keywords for adding statistics to the OUTPUT data set:

LINP | XBETA
requests the linear predictor \( \eta = x' \beta \). For observations in which only the response variable is
missing, values of the linear predictor are computed even though these observations do not affect
the model fit. The default name is Xbeta.

LOWER
requests a lower Bayesian confidence band value for the predicted value. The default name is
Lower.

PEARSON | PEARS | RESCHI
requests the Pearson residual, \( (y - \mu) / \sqrt{V(\mu)} \), where \( \mu \) is the estimate of the predicted response
mean and \( V(\mu) \) is the response distribution variance function. The default name is Pearson.

PREDICTED | PRED | P
requests predicted values for the response variable. For observations in which only the response
variable is missing, the predicted values are computed even though these observations do not
affect the model fit. The default name is Pred.

RESIDUAL | RESID | R
requests the raw residual, \( y - \mu \), where \( \mu \) is the estimate of the predicted mean. The default
name is Residual.

STD
requests a standard error for the linear predictor. The default name is Std.

UPPER
requests an upper Bayesian confidence band value for the predicted value. The default name is
Upper.

You can specify the following options in the OUTPUT statement after the slash (/):

ALPHA=number
specifies the significance level for the construction of Bayesian confidence bands in the OUTPUT data
set. The confidence level is \( 1 - number \).

COMPONENT
requests componentwise statistics for all spline terms if LINP, LOWER, STD, or UPPER is specified
as a keyword.
PERFORMANCE Statement

```
PERFORMANCE < performance-options > ;
```

You can use the PERFORMANCE statement to control whether the procedure executes in single-machine or distributed mode. The default is single-machine mode.

You can also use this statement to define performance parameters for multithreaded and distributed computing, and you can request details about performance results.


WEIGHT Statement

```
WEIGHT variable ;
```

The `variable` in the WEIGHT statement is used as a weight to perform a weighted analysis of the data. Observations that have nonpositive or missing weights are not included in the analysis. If a WEIGHT statement is not included, then all observations that are used in the analysis are assigned a weight of 1.

Details: GAMPL Procedure

Missing Values

Any observation that has missing values for the response, frequency, weight, offset, or explanatory variables is excluded from the analysis; however, missing values are valid for response and explanatory variables if you specify the MISSING option in the CLASS statement. Observations that have a nonpositive weight or a frequency less than 1 are also excluded. For Poisson, negative binomial and Tweedie distributions, observations that have a negative response value are excluded. For gamma and inverse Gaussian distributions, observations that have a nonpositive response value are excluded.

The estimated linear predictor and the fitted means are not computed for any observation that has missing offset or explanatory variable values. However, if only the response value is missing, the linear predictor and the fitted means can be computed and output to a data set by using the OUTPUT statement.

Thin-Plate Regression Splines

The GAMPL procedure uses thin-plate regression splines (Wood 2003) to construct spline basis expansions. The thin-plate regression splines are based on thin-plate smoothing splines (Duchon 1976, 1977). Compared to thin-plate smoothing splines, thin-plate regression splines produce fewer basis expansions and thus make direct fitting of generalized additive models possible.
Thin-Plate Smoothing Splines

Consider the problem of estimating a smoothing function \( f \) of \( x \) with \( d \) covariates from \( n \) observations. The model assumes

\[
y_i = f(x_i) + \epsilon_i, \quad i = 1, \ldots, n
\]

Then the thin-plate smoothing splines estimate the smoothing function \( f \) by minimizing the penalized least squares function:

\[
\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda J_{m,d}(f)
\]

The penalty term \( \lambda J_{m,d}(f) \) includes the function that measures roughness on the \( f \) estimate:

\[
J_{m,d}(f) = \int \cdots \int \sum_{\alpha_1 + \cdots + \alpha_d = m} \frac{m!}{\alpha_1! \cdots \alpha_d!} \left( \frac{\partial^m f}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}} \right)^2 \, dx_1 \cdots dx_d
\]

The parameter \( m \) (which corresponds to the M= option for a spline effect) specifies how the penalty is applied to the function roughness. Function derivatives whose order is less than \( m \) are not penalized. The relation \( 2m > d \) must be satisfied.

The penalty term also includes the smoothing parameter \( \lambda \in [0, \infty) \), which controls the trade-off between the model’s fidelity to the data and the function smoothness of \( f \). When \( \lambda = 0 \), the function estimate corresponds to an interpolation. When \( \lambda \to \infty \), the function estimate becomes the least squares fit. By using the defined penalized least squares criterion and a fixed \( \lambda \) value, you can explicitly express the estimate of the smooth function \( f \) in the following form:

\[
f_\lambda(x) = \sum_{j=1}^{M} \theta_j \phi_j(x) + \sum_{i} \delta_i \eta_{m,d}(\|x - x_i\|)
\]

In the expression of \( f_\lambda(x) \), \( \delta_i \) and \( \theta_j \) are coefficients to be estimated. The functions \( \phi_j(x) \) correspond to unpenalized polynomials of \( x \) with degrees up to \( m - 1 \). The total number of these polynomials is \( M = \binom{m+d-1}{d} \). The function \( \eta_{m,d} \) models the extra nonlinearity besides the polynomials and is a function of the Euclidean distance \( r \) between any \( x \) value and an observed \( x_i \) value:

\[
\eta_{m,d}(r) = \begin{cases} 
\frac{(-1)^m + 1 + d/2}{2^m \pi^{d/2} (m-1)! (m-d/2)!} r^{2m-d} \log(r) & \text{if } d \text{ is even} \\
\frac{\Gamma(d/2-m)}{2^{m} \pi^{d/2} (m-1)!} r^{2m-d} & \text{if } d \text{ is odd}
\end{cases}
\]

Define the penalty matrix \( E \) such that each entry \( E_{ij} = \eta_{m,d}(\|x_i - x_j\|) \), let \( y \) be the vector of the response, let \( T \) be the matrix where each row is formed by \( \phi_j(x) \), and let \( \theta \) and \( \delta \) be vectors of coefficients \( \theta_j \) and \( \delta_i \).

Then you can obtain the function estimate \( f_\lambda \) from the following minimization problem:

\[
\min \|y - T\theta - E\delta\|^2 + \lambda \delta' E \delta \quad \text{subject to} \quad T' \delta = 0
\]

For more information about thin-plate smoothing splines, see Chapter 119, “The TPSPLINE Procedure.”
Low-Rank Approximation

Given the representation of the thin-plate smoothing spline, the estimate of \( f \) involves as many parameters as the number of unique data points. Solving \((\theta, \delta)\) with an optimum \( \lambda \) becomes difficult for large problems.

Because the matrix \( E \) is symmetric and nonnegative definite, the eigendecomposition can be taken as \( E = UDU' \), where \( D \) is the diagonal matrix of eigenvalues \( d_i \) of \( E \), and \( U \) is the matrix of eigenvectors that corresponds to \( d_i \). The truncated eigendecomposition forms \( \tilde{E}_k \), which is an approximation to \( E \) such that

\[
\tilde{E}_k = U_k D_k U_k'
\]

where \( D_k \) is a diagonal matrix that contains the \( k \) most extreme eigenvalues in descending order of absolute values: \( |\tilde{d}_1| > \cdots > |\tilde{d}_k| \). \( U_k \) is the matrix that is formed by columns of eigenvectors that correspond to the eigenvalues in \( D_k \).

The approximation \( \tilde{E}_k \) not only reduces the dimension from \( n \times n \) of \( E \) to \( n \times k \) but also is optimal in two senses. First, \( \tilde{E}_k \) minimizes the spectral norm \( \|E - F_k\|_2 \) between \( E \) and all rank \( k \) matrices \( F_k \). Second, \( \tilde{E}_k \) also minimizes the worst possible change that is introduced by the eigenspace truncation as defined by

\[
\max_{\delta \neq 0} \frac{\delta'(E - G_k)\delta}{\|\delta\|^2}
\]

where \( G_k \) is formed by any \( k \) eigenvalues and corresponding eigenvectors. For more information, see Wood (2003).

Now given \( E \approx \tilde{E}_k \) and \( \tilde{E}_k = U_k D_k U_k' \), and letting \( \delta_k = U_k' \delta \), the minimization problem becomes

\[
\min \|y - T\theta - U_k D_k \delta_k\|^2 + \lambda \delta_k' D_k \delta_k \quad \text{subject to} \quad T' U_k \delta_k = 0
\]

You can turn the constrained optimization problem into an unconstrained one by using any orthogonal column basis \( Z \). One way to form \( Z \) is via the QR decomposition of \( U_k' T \):

\[
U_k' T = [Q_1 Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}
\]

Let \( Z = Q_2 \). Then it is verified that

\[
T' U_k Z = R' Q_1' Q_2 = 0
\]

So for \( \delta_k \) such that \( T' U_k \delta_k = 0 \), it is true that \( \delta_k = Z \tilde{\delta} \). Now the problem becomes the unconstrained optimization,

\[
\min \|y - T\theta - U_k D_k Z \tilde{\delta}\|^2 + \lambda \tilde{\delta}' Z' D_k Z \tilde{\delta}
\]

Let

\[
\beta = \begin{bmatrix} \theta \\ \tilde{\delta} \end{bmatrix}, \quad X = [T : U_k D_k Z], \quad \text{and} \quad S = \begin{bmatrix} 0 & 0 \\ 0 & Z' D_k Z \end{bmatrix}
\]

The optimization is simplified as

\[
\min \|y - X \beta\|^2 + \lambda \beta' S \beta \quad \text{with respect to} \quad \beta
\]
Generalized Additive Models

Generalized Linear Models

All probability distributions that the GAMPL procedure fits are members of an exponential family of distributions. For the specification of an exponential family, see the section “Generalized Linear Models Theory” in Chapter 46, “The GENMOD Procedure.”

Table 44.10 lists and defines some common notation that is used in the context of generalized linear models and generalized additive models.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \ell )</td>
<td>Log-likelihood</td>
</tr>
<tr>
<td>( \ell_p )</td>
<td>Penalized log-likelihood</td>
</tr>
<tr>
<td>( D )</td>
<td>Deviance</td>
</tr>
<tr>
<td>( D_p )</td>
<td>Penalized deviance</td>
</tr>
<tr>
<td>( g )</td>
<td>Link function</td>
</tr>
<tr>
<td>( g^{-1} )</td>
<td>Inverse link function</td>
</tr>
<tr>
<td>( \mu )</td>
<td>Response mean ( \mu = g^{-1}(\eta) )</td>
</tr>
<tr>
<td>( \eta )</td>
<td>Linear predictor ( \eta = X\beta )</td>
</tr>
<tr>
<td>( \phi )</td>
<td>Dispersion parameter</td>
</tr>
<tr>
<td>( z )</td>
<td>Column of adjusted response variable</td>
</tr>
<tr>
<td>( v )</td>
<td>Column of response variance</td>
</tr>
<tr>
<td>( \omega_i )</td>
<td>Prior weight for each observation</td>
</tr>
<tr>
<td>( w_i )</td>
<td>Adjusted weight for each observation</td>
</tr>
<tr>
<td>( W )</td>
<td>Diagonal matrix of adjusted weights</td>
</tr>
</tbody>
</table>

The GAMPL procedure supports the following distributions: binary, binomial, gamma, inverse Gaussian, negative binomial, normal (Gaussian), Poisson and Tweedie.

For forms of log-likelihood functions for each of the probability distributions, see the section “Log-Likelihood Functions” in Chapter 46, “The GENMOD Procedure.” For forms of the deviance for each of the probability distributions, see the section “Goodness of Fit” in Chapter 46, “The GENMOD Procedure.”

Generalized Additive Models

Generalized additive models are extensions of generalized linear models (Nelder and Wedderburn 1972). For each observation that has a response \( Y_i \) and a row vector of the model matrix \( x_i \), both generalized additive models and generalized linear models assume the model additivity

\[
g(\mu_i) = f_1(x_{i1}) + \cdots + f_p(x_{ip})
\]

where \( \mu_i = E(Y_i) \) and \( Y_i \) is independently distributed in some exponential family. Generalized linear models further assume model linearity by \( f_j(x_{ij}) = x_{ij} \beta_j \) for \( j = 1, \ldots, p \). Generalized additive models relax the linearity assumption by allowing some smoothing functions \( f_j \) to characterize the dependency. The GAMPL procedure constructs the smoothing functions by using thin-plate regression splines. For more
information about generalized additive models and other type of smoothing functions, see Chapter 43, “The GAM Procedure.”

Consider a generalized additive model that has some linear terms $X_L$ with coefficients $\beta_L$ and $p$ smoothing functions $f_j$. Each smoothing function can be constructed by thin-plate regression splines with a smoothing parameter $\lambda_j$. Using the notations described in the section “Low-Rank Approximation” on page 3059, you can characterize each smoothing function by

$$\beta_j = \begin{bmatrix} \theta_j \\ \delta_j \end{bmatrix}, \quad X_j = [T_j : U_{kj} D_{kj} Z_j], \quad \text{and} \quad S_j = \begin{bmatrix} 0 & 0 \\ 0 & Z_j' D_{kj} Z_j \end{bmatrix}$$

Notice that each smoothing function representation contains a zero-degree polynomial that corresponds to a constant. Having multiple constant terms makes the smoothing functions unidentifiable. The solution is to include a global constant term (that is, the intercept) in the model and enforce the centering constraint to each smoothing function. You can write the constraint as

$$1' X_j \beta_j = 0$$

By using a similar approach as the linear constraint for thin-plate regression splines, you obtain the orthogonal column basis $V_j$ via the QR decomposition of $X_j' 1$ such that $1' X_j V_j = 0$. Each smoothing function can be reparameterized as $Q X_j V_j = X_j' V_j$. Let $X = [X_L : \tilde{X}_1 : \cdots : \tilde{X}_p]$ and $\beta' = [\beta_L' : \beta_1' : \cdots : \beta_p']$. Then the generalized additive model can be represented as $g(\mu) = X \beta$. The roughness penalty matrix is represented as a block diagonal matrix:

$$S_\lambda = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & \lambda_1 S_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_p S_p \end{bmatrix}$$

Then the roughness penalty is measured in the quadratic form $\beta' S_\lambda \beta$.

**Penalized Likelihood Estimation**

Given a fixed vector of smoothing parameters, $\lambda = [\lambda_1 \cdots \lambda_p]'$, you can fit the generalized additive models by the penalized likelihood estimation. In contrast to the maximum likelihood estimation, penalized likelihood estimation obtains an estimate for $\beta$ by maximizing the penalized log likelihood,

$$\ell_p(\beta) = \ell(\beta) - \frac{1}{2} \beta' S_\lambda \beta$$

Any optimization technique that you can use for maximum likelihood estimation can also be used for penalized likelihood estimation. If first-order derivatives are required for the optimization technique, you can compute the gradient as

$$\frac{\partial \ell_p}{\partial \beta} = \frac{\partial \ell}{\partial \beta} - S_\lambda \beta$$
If second-order derivatives are required for the optimization technique, you can compute the Hessian as
\[
\frac{\partial^2 \ell_p}{\partial \beta \partial \beta'} = \frac{\partial^2 \ell}{\partial \beta \partial \beta'} - S_\lambda
\]

In the gradient and Hessian forms, \( \partial \ell / \partial \beta \) and \( \partial^2 \ell / (\partial \beta \partial \beta') \) are the corresponding gradient and Hessian, respectively, for the log-likelihood for generalized linear models. For more information about optimization techniques, see the section “Choosing an Optimization Technique” on page 3068.

**Model Evaluation Criteria**

Given a fixed set of smoothing parameters \( \lambda \) in which each \( \lambda_i \) controls the smoothness of each spline term, you can fit a generalized additive model by the penalized likelihood estimation. There are infinitely many sets of smoothing parameters. In order to search optimum models, some model evaluation criteria need to be defined to quantify the model goodness-of-fit. The GAMPL procedure uses the following model evaluation criteria:

- generalized cross validation (GCV), \( V_g \) (Craven and Wahba 1979)
- unbiased risk estimator (UBRE), \( V_u \) (Craven and Wahba 1979)
- generalized approximate cross validation (GACV), \( V_a \) (Xiang and Wahba 1996)

Consider the optimization problem
\[
\min_{\beta} (y - X\beta)'(y - X\beta) + \beta'S_\lambda \beta
\]
with respect to \( \beta \)

The parameter estimate for \( \beta \) can be represented as
\[
\hat{\beta} = (X'X + S_\lambda)^{-1}X'y
\]

And the smoothing matrix (also called the influence matrix or hat matrix) is thus represented as
\[
H_\lambda = X(X'X + S_\lambda)^{-1}X'
\]

With the defined smoothing matrix, you can form the model evaluation criteria as follows:

\[
V_g(\lambda) = \frac{n \| y - H_\lambda y \|^2}{(\text{tr}(I - yH_\lambda))^2}
\]

\[
V_u(\lambda) = \frac{1}{n} \| y - H_\lambda y \|^2 - \frac{2}{n} \sigma^2 \text{tr}(I - yH_\lambda) + \sigma^2
\]

\[
V_a(\lambda) = \frac{1}{n} \| y - H_\lambda y \|^2 \left( 1 + 2\gamma \frac{\text{tr}(H_\lambda)}{\text{tr}(I - H_\lambda)} \right)
\]

In the equations, \( \gamma \geq 1 \) (which corresponds to the GAMMA= suboption of the CRITERION= option) is the tuning parameter that is sometimes used to enforce smoother models.

The GAMPL procedure uses fitting algorithms that involve minimizing the model evaluation criterion with respect to unknown smoothing parameters \( \lambda \).
Fitting Algorithms

For models that assume a normally distributed response variable, you can minimize the model evaluation criteria directly by searching for optimal smoothing parameters. For models that have nonnormal distributions, you cannot use the model evaluation criteria directly because the involved statistics keep changing between iterations. The GAMPL procedure enables you to use either of two fitting approaches to search for optimum models: the outer iteration method and the performance iteration method. The outer iteration method modifies the model evaluation criteria so that a global objective function can be minimized in order to find the best smoothing parameters. The performance iteration method minimizes a series of objective functions in an iterative fashion and then obtains the optimum smoothing parameters at convergence. For large data sets, the performance iteration method usually converges faster than the outer iteration method.

Outer Iteration

The outer iteration method is outlined in Wood (2006). The method uses modified model evaluation criteria, which are defined as follows:

\[
V_o^\lambda = \frac{n D_\lambda(\mu)}{(n - y \text{tr}(\mathbf{H}_\lambda))^2}
\]

\[
V_d^\lambda = \frac{D_\lambda(\mu)}{n} - 2 \sigma^2 \text{tr}(\mathbf{I} - \gamma \mathbf{H}_\lambda) + \sigma^2
\]

\[
V_a^\lambda = \frac{D_\lambda(\mu)}{n} + \frac{2y \text{tr}(\mathbf{H}_\lambda) P_\lambda}{\text{ntr}(\mathbf{I} - \mathbf{H}_\lambda)}
\]

By replacing \(\|y - \mathbf{H}_\lambda y\|^2\) with model deviance \(D_\lambda(\mu)\), the modified model evaluation criteria relate to the smoothing parameter \(\lambda\) in a direct way such that the analytic gradient and Hessian are available in explicit forms. The Pearson’s statistic \(P_\lambda\) is used in the GACV criterion \(V_a^\lambda(\lambda)\) (Wood 2008). The algorithm for the outer iteration is thus as follows:

1. Initialize smoothing parameters by taking one step of performance iteration based on adjusted response and adjusted weight except for spline terms with initial values that are specified in the INITSMOOTH= option.

2. Search for the best smoothing parameters by minimizing the modified model evaluation criteria. The optimization process stops when any of the convergence criteria that are specified in the SMOOTHOPTIONS option is met. At each optimization step:
   a) Initialize by setting initial regression parameters \(\beta = \{g(\hat{y}), 0, \ldots, 0\}'\). Set the initial dispersion parameter if necessary.
   b) Search for the best regression parameters \(\beta\) by minimizing the penalized deviance \(D_p\) (or maximizing the penalized likelihood \(\ell_p\) for negative binomial distribution). The optimization process stops when any of the convergence criteria that are specified in the PLIKEOPTIONS option is met.
   c) At convergence, evaluate derivatives of the model evaluation criteria with respect to \(\lambda\) by using \(\partial D_p/\partial \beta\), \(\partial^2 D_p/(\partial \beta \partial \beta')\), \(\partial \beta / \partial \lambda_j\), and \(\partial^2 \beta / (\partial \lambda_j \partial \lambda_k)\).
Step 2b usually converges quickly because it is essentially penalized likelihood estimation given that
\[ D_p = 2\phi(\ell_{\text{max}} - \ell) + \beta S_\lambda \beta'. \]
Step 2c contains involved computation by using the chain rule of derivatives. For more information about computing derivatives of \( V^0_g \) and \( V^0_u \), see Wood (2008, 2011).

**Performance Iteration**

The performance iteration method is proposed by Gu and Wahba (1991). Wood (2004) modifies and stabilizes the algorithm for fitting generalized additive models. The algorithm for the performance iteration method is as follows:

1. Initialize smoothing parameters \( \lambda = \{1, \ldots, 1\} \), except for spline terms whose initial values are specified in the \texttt{INITSMOOTH=} option. Set the initial regression parameters \( \beta = \{g(\hat{y}), 0, \ldots, 0\}' \). Set the initial dispersion parameter if necessary.

2. Repeat the following steps until any of these three conditions is met: (1) the absolute function change in penalized likelihood is sufficiently small; (2) the absolute relative function change in penalized likelihood is sufficiently small; (3) the number of iterations exceeds the maximum iteration limit.
   a) Form the adjusted response and adjusted weight from \( \mu = g^{-1}(\eta) \). For each observation,
      \[ z_i = \eta_i + (y_i - \mu_i)/\mu'_i, \quad w_i = \omega_i \mu_i^2/y_i \]
   b) Search for the best smoothing parameters for the current iteration based on valid adjusted response values and adjusted weight values.
   c) Use the smoothing parameters to construct the linear predictor and the predicted means.
   d) Obtain an estimate for the dispersion parameter if necessary.

In step 2b, you can use different optimization techniques to search for the best smoothing parameters. The Newton-Raphson optimization is efficient in finding the optimum \( \lambda \) where the first- and second-order derivatives are available. For more information about computing derivatives of \( V^0_g \) and \( V^0_u \) with respect to \( \lambda \), see Wood (2004).

**Degrees of Freedom**

Let \( W \) be the adjusted weight matrix at convergence, and let \( S_\lambda \) be the roughness penalty matrix with selected smoothing parameters. The degrees of freedom matrix is defined as in Wood (2006):

\[ F = (X'WX + S_\lambda)^{-1}X'WX \]

Given the adjusted response \( z \), the parameter estimate is shown to be \( \tilde{\beta} = (X'WX)^{-1}X'Wz \) for the model without penalization, and the parameter estimate is \( \hat{\beta} = (X'WX + S_\lambda)^{-1}X'Wz = F\tilde{\beta} \) with penalization. \( F \) is thus the matrix that projects or maps the unpenalized parameter estimates to the penalized ones.

The model degrees of freedom is given as

\[ df = \text{tr}(F) \]
And the degrees of freedom for error is given as
\[ df_r = n - 2df + \text{tr}(FF) \]

For the \( j \)th spline term, the degrees of freedom for the component is defined to be the trace of the submatrix of \( F \) that corresponds to parameter estimates \( \beta_j \):
\[ df_j = \text{tr}(F_j) \]

The degrees of freedom for the smoothing component test of the \( j \)th term is defined similarly as
\[ df'_j = 2df_j - \text{tr}((FF)_j) \]

---

**Model Inference**

Wahba (1983) proposes a Bayesian covariance matrix for parameter estimates \( \beta \) by interpreting a smoothing spline as a posterior mean. Nychka (1988) shows that the derived Bayesian posterior confidence limits work well from frequentist viewpoints. The Bayesian posterior covariance matrix for the parameters is
\[ V_\beta = (X'WX + S_\lambda)^{-1} \sigma^2 \]

The posterior distribution for \( \beta \) is thus
\[ \beta | y \sim N(\hat{\beta}, V_\beta) \]

For a particular point whose design row is vector \( x \), the prediction is \( x\hat{\beta} \) and the standard error is \( \sqrt{xV_\beta x'} \). The Bayesian posterior confidence limits are thus
\[ \left( x\hat{\beta} \pm z_{\alpha/2} \sqrt{xV_\beta x'} \right) \]

where \( z_{\alpha/2} \) is the \( 1 - \alpha/2 \) quantile of the standard normal distribution.

For the \( j \)th spline term, the prediction for the component is formed by \( x_j\hat{\beta} \), where \( x_j \) is a row vector of zeros except for columns that correspond to basis expansions of the \( j \)th spline term. And the standard error for the component is \( \sqrt{x_jV_\beta x'_j} \).

---

**Dispersion Parameter**

Some distribution families (Gaussian, gamma, inverse Gaussian, negative binomial, and Tweedie) have a dispersion parameter that you can specify in the DISPERSON= option in the MODEL statement or you can estimate from the data. The following three suboptions for the SCALE= option in the MODEL statement correspond to three ways to estimate the dispersion parameter:

**DEVIANCE**

estimates the dispersion parameter by the deviance, given the regression parameter estimates:
\[ \hat{\phi} = \frac{\sum_i D_i(y_i, \mu_i)}{n - df} \]
MLE estimates the dispersion parameter by maximizing the penalized likelihood, given the regression parameter estimates:

\[ \hat{\phi} = \arg\max_{\phi} \ell_p(\hat{\beta}, \phi) \]

The MLE option is the only option that you can use to estimate the dispersion parameter for the negative binomial and Tweedie distributions.

PEARSON estimates the dispersion parameter by Pearson’s statistic, given the regression parameter estimates:

\[ \hat{\phi} = \frac{\sum_i \omega_i(y_i - \mu_i)^2 / v_i}{n - df} \]

If the dispersion parameter is estimated, it contributes one additional degree of freedom to the fitted model.

**Tests for Smoothing Components**

The GAMPL procedure performs a smoothing component test on the null hypotheses \( f_j = 0 \) for the \( j \)th component. In contrast to the analysis of deviance that is used in PROC GAM (which tests existence of nonlinearity for each smoothing component), the smoothing component test used in PROC GAMPL tests for the existence of a contribution for each smoothing component.

The hypothesis test is based on the Wald statistic. Define \( X_j \) as the matrix of all zeros except for columns that correspond to basis expansions of the \( j \)th spline term. Then the column vector of predictions is \( \hat{f}_j = X_j \hat{\beta} \), and the covariance matrix for the predictions is \( V_j = X_j V \beta X_j' \). The Wald statistic for testing is

\[ T_r = \hat{f}_j V_j^{-1} \hat{f}_j = \hat{\beta}' X_j' (X_j V \beta X_j')^{-1} X_j \hat{\beta} \]

where \( V_j^{-1} \) is the rank-\( r \) pseudo-inverse of \( V_j \). If \( R_j \) is the Cholesky root for \( X_j' X_j \) such that \( R_j' R_j = X_j' X_j \), then the test statistic can be written as

\[ T_r = \hat{\beta}' R_j' (R_j V \beta R_j')^{-1} R_j \hat{\beta} \]

Wood (2012) proposes using the \( df_j \) degrees of freedom for test (which is defined in the section “Degrees of Freedom” on page 3064) as the rank \( r \). Because spline terms in fitted models often have noninteger degrees of freedom, the GAMPL procedure uses a rounded value of \( df_j \) as the rank:

\[ r = \begin{cases} \lfloor df_j \rfloor & \text{if } df_j - \lfloor df_j \rfloor \leq 0.05 \text{ or } df_j < 1 \\ \lfloor df_j \rfloor & \text{otherwise} \end{cases} \]

Let \( K \) be a symmetric and nonnegative definite matrix, and let its eigenvalues be sorted as \( d_1 > d_2 > \cdots \); then the rank-\( r \) pseudo-inverse of \( K \) is formed by

\[ K^{-1} = U_k \begin{bmatrix} d_1^{-1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_r^{-1} \end{bmatrix} U_k' \]

where \( U_k \) are formed by columns of eigenvectors that correspond to the \( r \) eigenvalues.
Under the null hypothesis, the Wald statistic $T_r$ approximately follows the chi-square distribution $T_r \sim \chi^2_r$. For an observed test statistic $t_r$, the $p$-value for rejecting the null hypothesis is computed as $P(\chi^2_r > t_r)$ if the dispersion parameter is constant, or $P(F_{r,\text{df}_r} > t_r)$ with $\text{df}_r$ error degrees of freedom if the dispersion parameter is estimated.

Be cautious when you interpret the results of the smoothing component test because $p$-values are computed by approximation and the test does not take the smoothing parameter selection process into account.

---

**Computational Method: Multithreading**

Threading is the organization of computational work into multiple tasks (processing units that can be scheduled by the operating system). Each task is associated with a thread. Multithreading is the concurrent execution of threads. When multithreading is possible, you can realize substantial performance gains compared to the performance you get from sequential (single-threaded) execution.

The number of threads that the GAMPL procedure spawns is determined by the number of CPUs on a machine and can be controlled in the following ways:

- You can specify the number of CPUs in the CPUCOUNT= SAS system option. For example, if you specify the following statement, the GAMPL procedure determines threading as if it were executing on a system that had four CPUs, regardless of the actual CPU count:

  ```
  options cpucount=4;
  ```

- You can specify the NTHREADS= option in the PERFORMANCE statement to control the number of threads. This specification overrides the CPUCOUNT= system option. Specify NTHREADS=1 to force single-threaded execution.

The GAMPL procedure allocates one thread per CPU by default.

The tasks that are multithreaded by the GAMPL procedure are primarily defined by dividing the data that are processed on a single machine among the threads—that is, the GAMPL procedure implements multithreading through a data-parallel model. For example, if the input data set has 1,000 observations and PROC GAMPL is running with four threads, then 250 observations are associated with each thread. All operations that require access to the data are then multithreaded. These operations include the following:

- variable levelization
- effect levelization
- formation of the initial crossproducts matrix
- truncated eigendecomposition
- formation of spline basis expansions
- objective function calculation
- gradient calculation
Choosing an Optimization Technique

First- or Second-Order Techniques

The factors that affect how you choose an optimization technique for a particular problem are complex. Although the default technique works well for most problems, you might occasionally benefit from trying several different techniques.

For many optimization problems, computing the gradient takes more computer time than computing the function value. Computing the Hessian sometimes takes much more computer time and memory than computing the gradient, especially when there are many decision variables. Unfortunately, optimization techniques that do not use some kind of Hessian approximation usually require many more iterations than techniques that use a Hessian matrix; as a result, the total run time of these techniques is often longer. Techniques that do not use the Hessian also tend to be less reliable. For example, they can terminate more easily at stationary points than at global optima.

Table 44.11 shows which derivatives are required for each optimization technique.

<table>
<thead>
<tr>
<th>Technique</th>
<th>First-Order</th>
<th>Second-Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUREG</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>NEWRAP</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>NRRIDG</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>QUANEW</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>DBLDOG</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>CONGRA</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>NMSIMP</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

The second-order derivative techniques (TRUREG, NEWRAP, and NRRIDG) are best for small problems for which the Hessian matrix is not expensive to compute. Sometimes the NRRIDG technique can be faster than the TRUREG technique, but TRUREG can be more stable. The NRRIDG technique requires only one matrix with \( p(p + 1)/2 \) double words, where \( p \) denotes the number of parameters in the optimization. TRUREG and NEWRAP require two such matrices.

The QUANEW and DBLDOG first-order derivative techniques are best for medium-sized problems for which the objective function and the gradient can be evaluated much faster than the Hessian. In general, the QUANEW and DBLDOG techniques require more iterations than TRUREG, NRRIDG, and NEWRAP, but...
Choosing an Optimization Technique

Choosing an Optimization Technique

Each iteration can be much faster. The QUANEW and DBLDOG techniques require only the gradient to update an approximate Hessian, and they require slightly less memory than TRUREG or NEWRAP.

The CONGRA first-order derivative technique is best for large problems for which the objective function and the gradient can be computed much faster than the Hessian and for which too much memory is required to store the (approximate) Hessian. In general, the CONGRA technique requires more iterations than QUANEW or DBLDOG, but each iteration can be much faster. Because CONGRA requires only a factor of $p$ double-word memory, many large applications can be solved only by CONGRA.

The no-derivative technique NMSIMP is best for small problems for which derivatives are not continuous or are very difficult to compute.

Each optimization technique uses one or more convergence criteria that determine when it has converged. A technique is considered to have converged when any one of the convergence criteria is satisfied. For example, under the default settings, the QUANEW technique converges if the value of the ABSGCONV= option is less than 1E–5, the value of the FCONV= option is less than $2 \times \epsilon$, or the value of the GCONV= option is less than 1E–8.

By default, the GAMPL procedure applies the NEWRAP technique to optimization for selecting smoothing parameters by using the performance iteration method. For the outer iteration method, the GAMPL procedure applies the QUANEW technique for selecting smoothing parameters.

Technique Descriptions

The following subsections provide details about each optimization technique and follow the same order as Table 44.11.

Trust Region Optimization (TRUREG)
The trust region technique (TRUREG) uses the gradient $g(\psi^{(k)})$ and the Hessian matrix $H(\psi^{(k)})$; thus, it requires that the objective function $f(\psi)$ have continuous first- and second-order derivatives inside the feasible region.

The trust region technique iteratively optimizes a quadratic approximation to the nonlinear objective function within a hyperelliptic trust region that has radius $\Delta$ and constrains the step size that corresponds to the quality of the quadratic approximation. The trust region technique that PROC GAMPL uses is based on Dennis, Gay, and Welsch (1981); Gay (1983); Moré and Sorensen (1983).

The trust region technique performs well for small- to medium-sized problems, and it does not need many function, gradient, and Hessian calls. However, if the computation of the Hessian matrix is computationally expensive, either the QUANEW technique or the CONGRA technique might be more efficient.

Newton-Raphson Optimization with Line Search (NEWRAP)
The Newton-Raphson optimization with line search (NEWRAP) technique uses the gradient $g(\psi^{(k)})$ and the Hessian matrix $H(\psi^{(k)})$; thus, it requires that the objective function have continuous first- and second-order derivatives inside the feasible region.

If second-order derivatives are computed efficiently and precisely, the NEWRAP technique can perform well for medium-sized to large problems, and it does not need many function, gradient, and Hessian calls.

This technique uses a pure Newton step when the Hessian is positive-definite and when the Newton step reduces the value of the objective function successfully. Otherwise, a combination of ridging and line search
Chapter 44: The GAMPL Procedure

is performed to compute successful steps. If the Hessian is not positive-definite, a multiple of the identity matrix is added to the Hessian matrix to make it positive-definite (Eskow and Schnabel 1991).

In each iteration, a line search is performed along the search direction to find an approximate optimum of the objective function. The line-search technique uses quadratic interpolation and cubic extrapolation.

**Newton-Raphson Ridge Optimization (NRRIDG)**

The Newton-Raphson ridge optimization (NRRIDG) technique uses the gradient \( g(\psi^{(k)}) \) and the Hessian matrix \( H(\psi^{(k)}) \); thus, it requires that the objective function have continuous first- and second-order derivatives inside the feasible region.

This technique uses a pure Newton step when the Hessian is positive-definite and when the Newton step reduces the value of the objective function successfully. If at least one of these two conditions is not satisfied, a multiple of the identity matrix is added to the Hessian matrix.

Because the NRRIDG technique uses an orthogonal decomposition of the approximate Hessian, each iteration of NRRIDG can be slower than an iteration of the NEWRAP technique, which works with a Cholesky decomposition. However, NRRIDG usually requires fewer iterations than NEWRAP.

The NRRIDG technique performs well for small- to medium-sized problems, and it does not require many function, gradient, and Hessian calls. However, if the computation of the Hessian matrix is computationally expensive, either the QUANEW technique or the CONGRA technique might be more efficient.

**Dual Quasi-Newton Optimization (QUANEW)**

The dual quasi-Newton (QUANEW) technique uses the gradient \( g(\psi^{(k)}) \), and it does not need to compute second-order derivatives, because they are approximated. It works well for medium-sized to moderately large optimization problems, where the objective function and the gradient can be computed much faster than the Hessian. However, the QUANEW technique requires more iterations in general than the TRUREG, NEWRAP, and NRRIDG techniques, which compute second-order derivatives. The QUANEW technique provides an appropriate balance between the speed and stability that are required for most generalized linear model applications.

The QUANEW technique that PROC GAMPL uses is the dual quasi-Newton technique, which updates the Cholesky factor of an approximate Hessian.

In each iteration, a line search is performed along the search direction to find an approximate optimum. The line-search technique uses quadratic interpolation and cubic extrapolation to obtain a step size \( \alpha \) that satisfies the Goldstein conditions (Fletcher 1987). One of the Goldstein conditions can be violated if the feasible region defines an upper limit of the step size. Violating the left-side Goldstein condition can affect the positive-definiteness of the quasi-Newton update. In that case, either the update is skipped or the iterations are restarted by using an identity matrix, resulting in the steepest descent or ascent search direction.

**Double-Dogleg Optimization (DBLDOG)**

The double-dogleg optimization (DBLDOG) technique combines the ideas of the quasi-Newton and trust region techniques. In each iteration, the double-dogleg technique computes the step \( s^{(k)} \) as the linear combination of the steepest descent or ascent search direction \( s_1^{(k)} \) and a quasi-Newton search direction \( s_2^{(k)} \):

\[
s^{(k)} = \alpha_1 s_1^{(k)} + \alpha_2 s_2^{(k)}
\]

The step must remain within a prespecified trust region radius (Fletcher 1987, p. 107). Thus, the DBLDOG subroutine uses the dual quasi-Newton update but does not perform a line search.
The double-dogleg optimization technique works well for medium-sized to moderately large optimization problems, where the objective function and the gradient can be computed much faster than the Hessian. The implementation is based on Dennis and Mei (1979) and Gay (1983), but it is extended for dealing with boundary and linear constraints. The DBLDOG technique generally requires more iterations than the TRUREG, NEWRAP, and NRRIDG techniques, which require second-order derivatives; however, each of the DBLDOG iterations is computationally cheap. Furthermore, the DBLDOG technique requires only gradient calls for updating of the Cholesky factor of an approximate Hessian.

**Conjugate Gradient Optimization (CONGRA)**

Second-order derivatives are not required by the CONGRA technique and are not even approximated. The CONGRA technique can be expensive in function and gradient calls, but it requires only $O(p)$ memory for unconstrained optimization. In general, the technique must perform many iterations to obtain a precise solution, but each of the CONGRA iterations is computationally cheap.

The CONGRA technique should be used for optimization problems that have large $p$. For the unconstrained or boundary-constrained case, the CONGRA technique requires only $O(p)$ bytes of working memory, whereas all other optimization techniques require order $O(p^2)$ bytes of working memory. During $p$ successive iterations that are uninterrupted by restarts or changes in the working set, the CONGRA technique computes a cycle of $p$ conjugate search directions. In each iteration, a line search is performed along the search direction to find an approximate optimum of the objective function. The line-search technique uses quadratic interpolation and cubic extrapolation to obtain a step size $\alpha$ that satisfies the Goldstein conditions. One of the Goldstein conditions can be violated if the feasible region defines an upper limit for the step size.

**Nelder-Mead Simplex Optimization (NMSIMP)**

The Nelder-Mead simplex technique does not use any derivatives and does not assume that the objective function has continuous derivatives. The objective function itself needs to be continuous. This technique is quite expensive in the number of function calls, and it might be unable to generate precise results for $p \gg 40$.

The original Nelder-Mead simplex technique is implemented and extended to boundary constraints. This technique does not compute the objective for infeasible points, but it changes the shape of the simplex and adapts to the nonlinearities of the objective function. This change contributes to an increased speed of convergence and uses a special termination criterion.

---

**Displayed Output**

The following sections describe the output that the GAMPL procedure produces by default. The output is organized into various tables, which are discussed in the order of their appearance.

**Performance Information**

The “Performance Information” table is produced by default. It displays information about the execution mode. For single-machine mode, the table displays the number of threads used. For distributed mode, the table displays the grid mode (symmetric or asymmetric), the number of compute nodes, and the number of threads per node.

If you specify the DETAILS option in the PERFORMANCE statement, PROC GAMPL also produces a “Timing” table, which displays elapsed times (absolute and relative) for the main tasks of the procedure.
Data Access Information

The “Data Access Information” table is produced by default. For the input and output data sets, it displays the libref and data set name, the engine that was used to access the data, the role (input or output) of the data set, and the path that the data followed to reach the computation.

Model Information

The “Model Information” table displays basic information about the model, such as the response variable, frequency variable, link function, and model category that the GAMPL procedure determines based on your input and options. The “Model Information” table also displays the distribution of the data that PROC GAMPL assumes. For information about the supported response distributions, see the section “DISTRIBUTION=keyword” on page 3052.

Number of Observations

The “Number of Observations” table displays the number of observations that are read from the input data set and the number of observations that are used in the analysis. If a FREQ statement is present, the sum of the frequencies read and used is displayed. If the events/trials syntax is used, the number of events and trials is also displayed.

Response Profile

The “Response Profile” table displays the ordered values from which the GAMPL procedure determines the probability that is modeled as an event in binary models. For each response category level, the frequency that is used in the analysis is reported. You can affect the ordering of the response values by specifying response-options in the MODEL statement. For binary models, the note that follows the “Response Profile” table indicates which outcome is modeled as the event in binary models and which value serves as the reference category.

The “Response Profile” table is not produced for binomial (events/trials) data. You can find information about the number of events and trials in the “Number of Observations” table.

Class Level Information

The “Class Level Information” table lists the levels of every variable that is specified in the CLASS statement. You should check this information to make sure that the data are correct. You can adjust the order of the CLASS variable levels by specifying the ORDER= option in the CLASS statement. You can suppress the “Class Level Information” table completely or partially by specifying the NOCLPRINT= option in the PROC GAMPL statement.

If the classification variables use reference parameterization, the “Class Level Information” table also displays the reference value for each variable.

Specifications for Spline(spline-variables)

The “Specifications for Spline(spline-variables)” table displays basic information (such as number of variables, specified degrees of freedom, search range for the smoothing parameter, and so on) about how to construct a spline term that the GAMPL procedure uses to construct basis expansions and search for the smoothing parameter. PROC GAMPL generates the “Specifications for Spline(spline-variables)” table only when you specify the DETAILS option for a spline term.
**Optimization Iteration History**

For each iteration of the optimization, the “Iteration History” table displays the number of function evaluations (including gradient and Hessian evaluations), the value of the objective function, the change in the objective function from the previous iteration, and the absolute value of the largest (projected) gradient element. The objective function that PROC GAMPL uses in the optimization is normalized by default to enable comparisons across data sets that have different sampling intensity.

If you specify the ITDETAILS option in the PROC GAMPL statement, information about the parameter estimates and gradients in the course of the optimization is added to the “Iteration History” table.

For a parametric generalized linear model or for a generalized additive model that has fixed smoothing parameters, the “Iteration History” table displays information about regression parameter estimates and gradients. For a generalized additive model that has unknown smoothing parameters, the “Iteration History” table displays information about smoothing parameter estimates and gradients. If the performance iteration method is used, a column of performance iteration steps is added to the table.

**Convergence Status**

The convergence status table is a small ODS table that follows the “Iteration History” table in the default output. In the listing, this table appears as a message that indicates whether the optimization succeeded and which convergence criterion was met. If the optimization fails, the message indicates the reason for the failure. If you save the convergence status table to an output data set, a numeric Status variable is added that enables you to programmatically assess convergence. The values of the Status variable encode the following:

0  Convergence was achieved, or an optimization was not performed because TECHNIQUE=NONE was specified.
1  The objective function could not be improved.
2  Convergence was not achieved because of a user interrupt or because a limit (such as the maximum number of iterations or the maximum number of function evaluations) was reached. To modify these limits, see the MAXITER=, MAXFUNC=, and MAXTIME= options in the PROC GAMPL statement.
3  Optimization failed to converge because function or derivative evaluations failed at the starting values or during the iterations or because a feasible point that satisfies the parameter constraints could not be found in the parameter space.

**Fit Statistics**

The “Fit Statistics” table displays a variety of likelihood-based measures of fit in addition to the model roughness measurement. All information criteria are presented in “smaller is better” form.

The calculation of the information criteria uses the following formulas, where \( df \) denotes the model degrees of freedom, \( f \) denotes the number of frequencies used, and \( \ell \) is the log likelihood that is evaluated at the converged estimates:

\[
\text{AIC} = -2\ell + 2df \\
\text{AICC} = \begin{cases} 
-2\ell + 2df f/(f - df - 1) & \text{when } f > df + 2 \\
-2\ell + 2df(df + 2) & \text{otherwise} 
\end{cases} \\
\text{BIC} = -2\ell + df \log(f)
\]

If no FREQ statement is specified, then \( f \) equals \( n \), the number of observations that are used.
Parameter Estimates

The “Parameter Estimates” table displays the regression parameter estimates, their estimated (asymptotic) standard errors, chi-square statistics, and $p$-values for the hypothesis that the parameter is 0.

Estimates for Smoothing Components

The “Estimates for Smoothing Components” table displays a summary of the fitted spline terms, including effective degrees of freedom, smoothing parameters, roughness penalty values, number of parameters, rank of penalty matrix, and number of knots.

Tests for Smoothing Components

The “Tests for Smoothing Components” table displays effective degrees of freedom, effective degrees of freedom for test, $F/\chi^2$ statistics, and $p$-values for rejecting the hypothesis that the smoothing component has zero contribution.

ODS Table Names

Each table that the GAMPL procedure creates has a name that is associated with it, and you must use this name to refer to the table when you use ODS statements. These names are listed in Table 44.12.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Statement</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassLevels</td>
<td>Level information from the CLASS statement</td>
<td>CLASS</td>
<td></td>
</tr>
<tr>
<td>ConvergenceStatus</td>
<td>Status of optimization at conclusion of optimization</td>
<td>Default output</td>
<td></td>
</tr>
<tr>
<td>DataAccessInfo</td>
<td>Information about modes of data access</td>
<td>Default output</td>
<td></td>
</tr>
<tr>
<td>FitStatistics</td>
<td>Fit statistics</td>
<td>Default output</td>
<td></td>
</tr>
<tr>
<td>LikelihoodHist</td>
<td>Iteration history for maximum likelihood estimation or penalized likelihood estimation</td>
<td>PROC GAMPL ITDETAILS</td>
<td></td>
</tr>
<tr>
<td>ModelInfo</td>
<td>Information about the modeling environment</td>
<td>Default output</td>
<td></td>
</tr>
<tr>
<td>NObs</td>
<td>Number of observations read and used, and number of events and trials, if applicable</td>
<td>Default output</td>
<td></td>
</tr>
<tr>
<td>ParameterEstimates</td>
<td>Solutions for the parameter estimates that are associated with effects in MODEL statements</td>
<td>Default output</td>
<td></td>
</tr>
<tr>
<td>PerformanceInfo</td>
<td>Information about the high-performance computing environment</td>
<td>Default output</td>
<td></td>
</tr>
</tbody>
</table>
Table 44.12  continued

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Statement</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResponseProfile</td>
<td>Response categories and the category that is modeled in models for binary</td>
<td>Default output</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and multinomial data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SmoothingEstimates</td>
<td>Information for spline terms after model fitting</td>
<td>Default output</td>
<td></td>
</tr>
<tr>
<td>SmoothingHist</td>
<td>Iteration history for smoothing parameter estimation</td>
<td>PROC GAMPL</td>
<td>ITDETAILS</td>
</tr>
<tr>
<td>SmoothingTests</td>
<td>Smoothing components test result</td>
<td>Default output</td>
<td></td>
</tr>
<tr>
<td>SplineDetails</td>
<td>Information about spline construction and smoothing parameter search</td>
<td>MODEL</td>
<td>SPLINE(\texttt{variables} / DETAILS)</td>
</tr>
<tr>
<td>Timing</td>
<td>Absolute and relative times for tasks performed by the procedure</td>
<td>PERFORMANCE</td>
<td>DETAILS</td>
</tr>
</tbody>
</table>

By referring to the names of such tables, you can use the ODS OUTPUT statement to place one or more of these tables in output data sets.

**ODS Graphics**

Statistical procedures use ODS Graphics to create graphs as part of their output. ODS Graphics is described in detail in Chapter 21, “Statistical Graphics Using ODS.”

Before you create graphs, ODS Graphics must be enabled (for example, by specifying the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section “Enabling and Disabling ODS Graphics” on page 615 in Chapter 21, “Statistical Graphics Using ODS.”

The overall appearance of graphs is controlled by ODS styles. Styles and other aspects of using ODS Graphics are discussed in the section “A Primer on ODS Statistical Graphics” on page 614 in Chapter 21, “Statistical Graphics Using ODS.”

When ODS Graphics is enabled, the GAMPL procedure by default produces plots of the partial predictions for each spline term in the model. Use the PLOTS option in the PROC GAMPL statement to control aspects of these plots.

PROC GAMPL assigns a name to each graph that it creates by using ODS. You can use these names to refer to the graphs when using ODS. The names are listed in Table 44.13.

Table 44.13  Graphs Produced by PROC GAMPL

<table>
<thead>
<tr>
<th>ODS Graph Name</th>
<th>Plot Description</th>
<th>PLOTS= Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>SmoothingComponentPanel</td>
<td>Panel of multiple prediction curves</td>
<td>Default COMPONENTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLOTS(UNPACK)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLOTS=COMPONENTS(UNPACK)</td>
</tr>
<tr>
<td>SmoothingComponentPlot</td>
<td>Unpacked prediction curves</td>
<td></td>
</tr>
</tbody>
</table>
By default, prediction plots for each spline component are displayed in panels that contain at most six plots. If you specify more than six smoothing components, multiple panels are used. Use the PLOTS(UNPACK) option in the PROC GAMPL statement to display these plots individually.

### Examples: GAMPL Procedure

#### Example 44.1: Scatter Plot Smoothing

This example shows how you can use PROC GAMPL to perform scatter plot smoothing.

The example uses the LIDAR data set (Ruppert, Wand, and Carroll 2003). This data set is used in many books and journals to illustrate different smoothing techniques. Scientists use a technique known as LIDAR (light detection and ranging), which uses laser reflections to detect chemical compounds in the atmosphere.

The following DATA step creates the data set Lidar:

```sas
title 'Scatter Plot Smoothing';
data Lidar;
   input Range LogRatio @@;
datalines;
390 -0.05035573 391 -0.06009706 393 -0.04190091 394 -0.0509847 396 -0.05991345 397 -0.02842392 399 -0.05958421 400 -0.03988881 402 -0.02939582 403 -0.03949445 405 -0.04764749 406 -0.04764749 408 -0.03123034 409 -0.03816584 411 -0.07562269 412 -0.05001751 414 -0.0457295 415 -0.07766966 417 -0.02460641 418 -0.07133184 ...
... more lines ...
702 -0.4716702 703 -0.7801088 705 -0.6668431 706 -0.5783479 708 -0.7874522 709 -0.6156956 711 -0.8967602 712 -0.7077379 714 -0.672567 715 -0.6218413 717 -0.8657611 718 -0.557754 720 -0.8026684
;
```

In this data set, `Range` records the distance that light travels before it is reflected back to the source. `LogRatio` is the logarithm of the ratio of light that is received from two laser sources. The objective is to use scatter plot smoothing to discover the nonlinear pattern in the data set. SAS provides different methods (for example, local regression) for scatter plot smoothing. You can perform scatter plot smoothing by using the SGPLOT procedure, as shown in the following statements:

```sas
proc sgplot data=Lidar;
   scatter x=Range y=LogRatio;
   loess x=Range y=LogRatio / nomarkers;
   pbspline x=Range y=LogRatio / nomarkers;
run;
```

Output 44.1.1 shows the scatter plot of `Range` and `LogRatio` and the smoothing curves that are fitted by local regression and penalized B-splines smoothing techniques.
Both scatter plot smoothing techniques show a significant nonlinear structure between \texttt{Range} and \texttt{LogRatio} that cannot be easily modeled by ordinary polynomials. You can also use the \texttt{GAMPL} procedure to perform scatter plot smoothing on this data set, as in the following statements:

```plaintext
proc gampl data=Lidar seed=12345;
   model LogRatio = spline(Range/details);
   output out=LidarOut pred=p;
run;
```

The “Specifications for Spline(Range)” table in \textbf{Output 44.1.2} displays the specifications for constructing the spline term for \texttt{Range}. The maximum degrees of freedom is 10, which sets the upper limit of effective degrees of freedom for the spline term to be 9 after one degree of freedom is absorbed in the intercept. The order of the derivative in the penalty is 2, which means the unpenalized portion of the spline term involves polynomials with degrees up to 2.
Output 44.1.2 Spline Specification

Scatter Plot Smoothing

The GAMPL Procedure

Specifications for Spline(Range)

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Variables</td>
<td>1</td>
</tr>
<tr>
<td>Rank of Penalty Approximation</td>
<td>10</td>
</tr>
<tr>
<td>Order of Derivative in the Penalty</td>
<td>2</td>
</tr>
<tr>
<td>Maximum Number of Knots</td>
<td>2000</td>
</tr>
</tbody>
</table>

The “Fit Statistics” table in Output 44.1.3 shows the summary statistics for the fitted model.

Output 44.1.3 Fit Statistics

<table>
<thead>
<tr>
<th>Fit Statistics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Penalized Log Likelihood</td>
<td>251.44124</td>
</tr>
<tr>
<td>Roughness Penalty</td>
<td>0.00000426</td>
</tr>
<tr>
<td>Effective Degrees of Freedom</td>
<td>9.99988</td>
</tr>
<tr>
<td>Effective Degrees of Freedom for Error</td>
<td>211.00000</td>
</tr>
<tr>
<td>AIC (smaller is better)</td>
<td>-482.88271</td>
</tr>
<tr>
<td>AICC (smaller is better)</td>
<td>-481.83512</td>
</tr>
<tr>
<td>BIC (smaller is better)</td>
<td>-448.90148</td>
</tr>
<tr>
<td>GCV (smaller is better)</td>
<td>0.00654</td>
</tr>
</tbody>
</table>

The “Estimates for Smoothing Components” table in Output 44.1.4 shows that the effective degrees of freedom for the spline term of Range is approximately 8 after the GCV criterion is optimized with respect to the smoothing parameter. The roughness penalty is small, suggesting that there is an important contribution from the penalized part of thin-plate regression splines beyond nonpenalized polynomials.

Output 44.1.4 Estimates for Smoothing Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Effective DF</th>
<th>Smoothing Parameter</th>
<th>Roughness Penalty</th>
<th>Number of Parameters</th>
<th>Rank of Penalty Matrix</th>
<th>Number of Knots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline(Range)</td>
<td>7.99988</td>
<td>1.0000</td>
<td>4.263E-6</td>
<td>9</td>
<td>10</td>
<td>221</td>
</tr>
</tbody>
</table>

Because the optimal model is obtained by searching in a functional space that is constrained by the maximum degrees of freedom for a spline term, you might wonder whether PROC GAMPL produces a much different model if you increase the value. The following statements fit another model in which the maximum degrees of freedom is increased to 20:

```plaintext
proc gampl data=Lidar seed=12345;
    model LogRatio = spline(Range/maxdf=20);
    output out=LidarOut2 pred=p2;
run;
```
Output 44.1.5 displays fit summary statistics for the second model. The model fit statistics from the second model are very close to the ones from the first model, indicating that the second model is not much different from the first model.

**Output 44.1.5  Fit Statistics**

Scatter Plot Smoothing

The GAMPL Procedure

<table>
<thead>
<tr>
<th>Fit Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Penalized Log Likelihood</td>
</tr>
<tr>
<td>Roughness Penalty</td>
</tr>
<tr>
<td>Effective Degrees of Freedom</td>
</tr>
<tr>
<td>Effective Degrees of Freedom for Error</td>
</tr>
<tr>
<td>AIC (smaller is better)</td>
</tr>
<tr>
<td>AICC (smaller is better)</td>
</tr>
<tr>
<td>BIC (smaller is better)</td>
</tr>
<tr>
<td>GCV (smaller is better)</td>
</tr>
</tbody>
</table>

Output 44.1.6 shows that the effective degrees of freedom for the spline term of Range is slightly larger than 8, which is understandable because increasing the maximum degrees of freedom expands the functional space for model searching. Functions in the expanded space can provide a better fit to the data, but they are also penalized more because the roughness penalty value for the second model is much larger than the one for the first model. This suggests that functions in the expanded space do not help much, given the nonlinear relationship between Range and LogRatio.

**Output 44.1.6  Estimates for Smoothing Components**

<table>
<thead>
<tr>
<th>Estimates for Smoothing Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
</tr>
<tr>
<td>Effective DF</td>
</tr>
<tr>
<td>Smoothing Parameter</td>
</tr>
<tr>
<td>Roughness Penalty</td>
</tr>
<tr>
<td>Number of Parameters</td>
</tr>
<tr>
<td>Rank of Penalty Matrix</td>
</tr>
<tr>
<td>Number of Knots</td>
</tr>
<tr>
<td>Spline(Range)</td>
</tr>
</tbody>
</table>

The two fitted models are all based on thin-plate regression splines, in which polynomials that have degrees higher than 2 are penalized. You might wonder whether allowing higher-order polynomials yields a much different model. The following statements fit a third spline model by penalizing polynomials that have degrees higher than 3:

```
proc gampl data=Lidar seed=12345;
 model LogRatio = spline(Range/m=3);
 output out=LidarOut3 pred=p3;
 run;
```

The fit summary statistics shown in Output 44.1.7 are close to the ones from the previous two models, albeit slightly smaller.
As shown in Output 44.1.8, the effective degrees of freedom for the spline term where polynomials with degrees less than 4 are allowed without penalization is 8. The roughness penalty is quite small compared to the previous two fits. This also suggests that there are important contributions from the penalized part of the thin-plate regression splines even after the nonpenalized polynomials are raised to order 3.

Output 44.1.8 Estimates for Smoothing Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Effective DF</th>
<th>Smoothing Parameter</th>
<th>Roughness Penalty</th>
<th>Number of Parameters</th>
<th>Rank of Penalty Matrix</th>
<th>Number of Knots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline(Range)</td>
<td>8.00000</td>
<td>1.0000</td>
<td>9.44E-9</td>
<td>9</td>
<td>10</td>
<td>221</td>
</tr>
</tbody>
</table>

The following statements use the DATA step to merge the predictions from the three scatter plot smoothing fits by PROC GAMPL and use the SGPLOT procedure to visualize them:

```plaintext
data LidarPred;
  merge Lidar LidarOut LidarOut2 LidarOut3;
run;

proc sgplot data=LidarPred;
  scatter x=Range y=LogRatio / markerattrs=GraphData1(size=7);
  series x=Range y=p / lineattrs =GraphData2(thickness=2)
                 legendlabel="Spline 1";
  series x=Range y=p2 / lineattrs =GraphData3(thickness=2)
                     legendlabel="Spline 2";
  series x=Range y=p3 / lineattrs =GraphData4(thickness=2)
                     legendlabel="Spline 3";
run;
```

Output 44.1.9 displays the scatter plot smoothing fits by PROC GAMPL under three different spline specifications.
Example 44.2: Nonparametric Logistic Regression

This example shows how you can use PROC GAMPL to build a nonparametric logistic regression model for a data set that contains a binary response and then use that model to classify observations.

The example uses the Pima Indians Diabetes data set, which can be obtained from the UCI Machine Learning Repository (Lichman 2013). It is extracted from a larger database that was originally owned by the National Institute of Diabetes and Digestive and Kidney Diseases. Data are for female patients who are at least 21 years old, are of Akimel O’otham (Pima Indian) heritage, and live near Phoenix, Arizona. The objective of this study is to investigate the relationship between a diabetes diagnosis and variables that represent physiological measurements and medical attributes. Some missing or invalid observations are removed from the analysis. The reduced data set contains 532 records. The following DATA step creates the data set DiabetesStudy:

title 'Diabetes Study';
data DiabetesStudy;
  input NPreg Glucose Pressure Triceps BMI Pedigree Age Diabetes Test@@;
datalines;
  6 148 72 35 33.6 0.627 50 1 1 1 85 66 29 26.6 0.351 31 0 1
  1 89 66 23 28.1 0.167 21 0 0 3 78 50 32 31 0.248 26 1 0
The data set contains nine variables, including the binary response variable Diabetes. Table 44.14 describes the variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPreg</td>
<td>Number of pregnancies</td>
</tr>
<tr>
<td>Glucose</td>
<td>Two-hour plasma glucose concentration in an oral glucose tolerance test</td>
</tr>
<tr>
<td>Pressure</td>
<td>Diastolic blood pressure (mm Hg)</td>
</tr>
<tr>
<td>Triceps</td>
<td>Triceps skin fold thickness (mm)</td>
</tr>
<tr>
<td>BMI</td>
<td>Body mass index (weight in kg/(height in m)^2)</td>
</tr>
<tr>
<td>Pedigree</td>
<td>Diabetes pedigree function</td>
</tr>
<tr>
<td>Age</td>
<td>Age (years)</td>
</tr>
<tr>
<td>Diabetes</td>
<td>0 if test negative for diabetes, 1 if test positive</td>
</tr>
<tr>
<td>Test</td>
<td>0 for training role, 1 for test</td>
</tr>
</tbody>
</table>

The Test variable splits the data set into training and test subsets. The training observations (whose Test value is 0) hold approximately 59.4% of the data. To build a model that is based on the training data and evaluate its performance by predicting the test data, you use the following statements to create a new variable, Result, whose value is the same as that of the Diabetes variable for a training observation and is missing for a test observation:

```sas
data DiabetesStudy;
  set DiabetesStudy;
  Result = Diabetes;
  if Test=1 then Result=.;
run;
```

As a starting point of your analysis, you can build a parametric logistic regression model on the training data and predict the test data. The following statements use PROC HPLOGISTIC to perform the analysis:

```sas
proc hplogistic data=DiabetesStudy;
  model Diabetes(event='1') = NPreg Glucose Pressure Triceps BMI Pedigree Age;
  partition role=Test(test='1' train='0');
run;
```

Output 44.2.1 shows the summary statistics from the parametric logistic regression model.
Output 44.2.1  Fit Statistics

Diabetes Study

The HPLOGISTIC Procedure

<table>
<thead>
<tr>
<th>Fit Statistics</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-2 Log Likelihood</td>
<td>295.79</td>
<td>176.90</td>
</tr>
<tr>
<td>AIC (smaller is better)</td>
<td>311.79</td>
<td>192.90</td>
</tr>
<tr>
<td>AICC (smaller is better)</td>
<td>312.24</td>
<td>193.65</td>
</tr>
<tr>
<td>BIC (smaller is better)</td>
<td>342.19</td>
<td>219.37</td>
</tr>
</tbody>
</table>

Output 44.2.2 shows fit statistics for both training and test subsets of the data, including the misclassification error for the test data.

Output 44.2.2  Partition Fit Statistics

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area under the ROC</td>
<td>0.8607</td>
<td>0.8547</td>
</tr>
<tr>
<td>Average Square Error</td>
<td>0.1452</td>
<td>0.1401</td>
</tr>
<tr>
<td>Hosmer-Lemeshow Test</td>
<td>0.9726</td>
<td>0.1487</td>
</tr>
<tr>
<td>Misclassification Error</td>
<td>0.2061</td>
<td>0.2178</td>
</tr>
<tr>
<td>R-Square</td>
<td>0.3438</td>
<td>0.2557</td>
</tr>
<tr>
<td>Max-rescaled R-Square</td>
<td>0.4695</td>
<td>0.3706</td>
</tr>
<tr>
<td>McFadden's R-Square</td>
<td>0.3197</td>
<td>0.2522</td>
</tr>
<tr>
<td>Mean Difference</td>
<td>0.3780</td>
<td>0.3605</td>
</tr>
<tr>
<td>Somers' D</td>
<td>0.7213</td>
<td>0.7093</td>
</tr>
<tr>
<td>True Negative Fraction</td>
<td>0.8702</td>
<td>0.8435</td>
</tr>
<tr>
<td>True Positive Fraction</td>
<td>0.6639</td>
<td>0.6182</td>
</tr>
</tbody>
</table>

The parametric logistic regression model is restricted in the sense that all variables affect the response in strictly linear fashion. If you are uncertain that a variable is an important factor and its contribution is linear in predicting the response, you might want to choose a nonparametric logistic regression model to fit the data. You can use PROC GAMPL to form a nonparametric model by including the spline transformation of each explanatory variable, as shown in the following statements:

```plaintext
proc gampl data=DiabetesStudy seed=12345;
  model Result(event='1') = spline(NPreg) spline(Glucose) spline(Pressure) spline(Triceps) spline(BMI) spline(Pedigree) spline(Age) / dist=binary;
run;
```

Because the response variable Result is binary, the DIST=BINARY option in the MODEL statement specifies a binary distribution for the response variable. By default, PROC GAMPL models the probability of the first ordered response category, which is a negative diabetes testing result in this case. The EVENT= option specifically requests that PROC GAMPL model the probability of positive diabetes testing results. The “Response Profile” table in Output 44.2.3 shows the frequencies of the response in both categories.
Output 44.2.3  Response Profile

Diabetes Study

The GAMPL Procedure

<table>
<thead>
<tr>
<th>Ordered Value</th>
<th>Result</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>208</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>122</td>
</tr>
</tbody>
</table>

You are modeling the probability that Result='1'.

Output 44.2.4 lists the summary statistics from the nonparametric logistic regression model, which include spline transformations of all variables.

Output 44.2.4  Fit Statistics

<table>
<thead>
<tr>
<th>Fit Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Penalized Log Likelihood: -130.55095</td>
</tr>
<tr>
<td>Roughness Penalty: 3.20011</td>
</tr>
<tr>
<td>Effective Degrees of Freedom: 25.85896</td>
</tr>
<tr>
<td>Effective Degrees of Freedom for Error: 302.97752</td>
</tr>
<tr>
<td>AIC (smaller is better): 309.61970</td>
</tr>
<tr>
<td>AICC (smaller is better): 314.20203</td>
</tr>
<tr>
<td>BIC (smaller is better): 407.86031</td>
</tr>
<tr>
<td>UBRE (smaller is better): 0.14225</td>
</tr>
</tbody>
</table>

The “Tests for Smoothing Components” table in Output 44.2.5 shows approximate tests results. Although some spline terms are significant, others are not. The null testing hypothesis is whether the total contribution from a variable is 0. So you can form a reduced model by removing those nonsignificant spline terms from the model. In this case, spline transformations for NPreg, Pressure, BMI, and Triceps are dropped from the model because their p-values are larger than the 0.1 nominal level.

Output 44.2.5  Tests for Smoothing Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Effective DF</th>
<th>DF for Test</th>
<th>Chi-Square</th>
<th>Pr &gt; ChiSq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline(NPreg)</td>
<td>1.00001</td>
<td>1</td>
<td>0.0758</td>
<td>0.7831</td>
</tr>
<tr>
<td>Spline(Glucose)</td>
<td>1.00000</td>
<td>1</td>
<td>38.4662</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Spline(Pressure)</td>
<td>7.80438</td>
<td>8</td>
<td>10.2605</td>
<td>0.2472</td>
</tr>
<tr>
<td>Spline(Triceps)</td>
<td>1.00000</td>
<td>1</td>
<td>0.6343</td>
<td>0.4258</td>
</tr>
<tr>
<td>Spline(BMI)</td>
<td>8.00000</td>
<td>8</td>
<td>11.8047</td>
<td>0.1601</td>
</tr>
<tr>
<td>Spline(Pedigree)</td>
<td>1.00002</td>
<td>1</td>
<td>8.5323</td>
<td>0.0035</td>
</tr>
<tr>
<td>Spline(Age)</td>
<td>5.05456</td>
<td>7</td>
<td>15.7671</td>
<td>0.0273</td>
</tr>
</tbody>
</table>
The following statements use PROC GAMPL to fit a reduced nonparametric logistic regression model. The OUTPUT statement requests predicted probabilities for both training and test data sets. The ID statement requests that the Diabetes and Test variables also be included in the output data set so that you can use them to identify test observations and compute misclassification errors.

```plaintext
ods graphics on;
proc gampl data=DiabetesStudy plots seed=12345;
  model Result(event='1') = spline(Glucose) spline(Pedigree) spline(Age) / dist=binary;
  output out=DiabetesStudyOut;
  id Diabetes Test;
run;
```

Output 44.2.6 shows the summary statistics from the reduced nonparametric logistic regression model. The values of the information criteria are better than of the parametric logistic regression model.

**Output 44.2.6 Fit Statistics**

<table>
<thead>
<tr>
<th>Fit Statistics</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Penalized Log Likelihood</td>
<td>-149.85765</td>
</tr>
<tr>
<td>Roughness Penalty</td>
<td>2.85613</td>
</tr>
<tr>
<td>Effective Degrees of Freedom</td>
<td>8.05242</td>
</tr>
<tr>
<td>Effective Degrees of Freedom for Error</td>
<td>320.61181</td>
</tr>
<tr>
<td>AIC (smaller is better)</td>
<td>312.96402</td>
</tr>
<tr>
<td>AICC (smaller is better)</td>
<td>313.41826</td>
</tr>
<tr>
<td>BIC (smaller is better)</td>
<td>343.55593</td>
</tr>
<tr>
<td>UBRE (smaller is better)</td>
<td>-0.00230</td>
</tr>
</tbody>
</table>

In the “Estimates for Smoothing Components” table in Output 44.2.7, PROC GAMPL reports that the effective degrees of freedom value for spline transformations of Glucose is quite close to 1. This suggests strictly linear form for Glucose. For Pedigree, the degrees of freedom value demonstrates a moderate amount of nonlinearity. For Age, the degrees of freedom value is much larger than 1. The measure suggests a nonlinear pattern in the dependency of the response on Age.

**Output 44.2.7 Estimates for Smoothing Components**

<table>
<thead>
<tr>
<th>Component</th>
<th>Effective DF</th>
<th>Smoothing Parameter</th>
<th>Roughness Penalty</th>
<th>Number of Parameters</th>
<th>Rank of Penalty Matrix</th>
<th>Number of Knots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline(Glucose)</td>
<td>1.00000</td>
<td>9.032E10</td>
<td>2.711E-7</td>
<td>9</td>
<td>10</td>
<td>110</td>
</tr>
<tr>
<td>Spline(Pedigree)</td>
<td>1.51071</td>
<td>0.4383</td>
<td>0.5086</td>
<td>9</td>
<td>10</td>
<td>283</td>
</tr>
<tr>
<td>Spline(Age)</td>
<td>4.54171</td>
<td>69.8810</td>
<td>2.3475</td>
<td>9</td>
<td>10</td>
<td>42</td>
</tr>
</tbody>
</table>

The “Tests for Smoothing Components” table in Output 44.2.8 shows that all spline transformations are significant in predicting diabetes testing results.
Output 44.2.8  Tests for Smoothing Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Effective DF</th>
<th>Effective DF for Test</th>
<th>Chi-Square</th>
<th>Pr &gt; ChiSq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline(Glucose)</td>
<td>1.00000</td>
<td>1</td>
<td>53.0363</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Spline(Pedigree)</td>
<td>1.51071</td>
<td>2</td>
<td>9.9354</td>
<td>0.0070</td>
</tr>
<tr>
<td>Spline(Age)</td>
<td>4.54171</td>
<td>6</td>
<td>23.0661</td>
<td>0.0008</td>
</tr>
</tbody>
</table>

The smoothing component panel (which is produced by the PLOTS option and is shown in Output 44.2.9) visualizes the spline transformations for the four variables in addition to 95% Bayesian curvewise confidence bands. For Glucose, the spline transformation is almost a straight line. For Pedigree, the spline transformation shows a slightly nonlinear trend. For Age, the dependency is obviously nonlinear.

Output 44.2.9  Smoothing Components Panel
Example 44.3: Nonparametric Negative Binomial Model for Mackerel Egg Density

The following statements compute the misclassification error on the test data set from the reduced nonparametric logistic regression model that PROC GAMPL produces:

```sas
data test;
  set DiabetesStudyOut(where=(Test=1));
  if ((Pred>0.5 & Diabetes=1) | (Pred<0.5 & Diabetes=0))
    then Error=0;
  else Error=1;
run;
proc freq data=test;
  tables Diabetes*Error/nocol norow;
run;
```

Output 44.2.10 shows the misclassification errors for observations in the test set and observations of each response category. The error is smaller than the error from the parametric logistic regression model.

**Output 44.2.10**  Crosstabulation Table for Test Set Prediction

<table>
<thead>
<tr>
<th></th>
<th>Table of Diabetes by Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Error</td>
</tr>
<tr>
<td>Frequency</td>
<td>0</td>
</tr>
<tr>
<td>Percent</td>
<td>64.36</td>
</tr>
<tr>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>17.33</td>
</tr>
<tr>
<td>Total</td>
<td>165</td>
</tr>
<tr>
<td></td>
<td>81.68</td>
</tr>
</tbody>
</table>

Example 44.3: Nonparametric Negative Binomial Model for Mackerel Egg Density

This example demonstrates how you can use PROC GAMPL to fit a nonparametric negative binomial model for a count data set that has overdispersions.

The example concerns a study of mackerel egg density. The data are a subset of the 1992 mackerel egg survey that was conducted over the Porcupine Bank west of Ireland. The survey took place in the peak spawning area. Scientists took samples by hauling a net up from deep sea to the sea surface. Then they counted the number of spawned mackerel eggs and used other geographic information to estimate the sizes and distributions of spawning stocks. The data set is used as an example in Bowman and Azzalini (1997).

The following DATA step creates the data set Mackerel. This data set contains 634 observations and five variables. The response variable Egg_Count is the number of mackerel eggs that are collected from each sampling net. Longitude and Latitude are the location values in degrees east and north, respectively, of each sample station. Net_Area is the area of the sampling net in square meters. Depth records the sea bed depth in meters at the sampling location, and Distance is the distance in geographic degrees from the sample location to the continental shelf edge.
title 'Mackerel Egg Density Study';
data Mackerel;
   input Egg_Count Longitude Latitude Net_Area Depth Distance;
datalines;
0  -4.65  44.57  0.242  4342  0.8395141177
0  -4.48  44.57  0.242  4334  0.8591926336
0  -4.3   44.57  0.242  4286  0.8930152895
1  -2.87  44.02  0.242  1438  0.3956408691
4  -2.07  44.02  0.242  166   0.0400088237
3  -2.13  44.02  0.242  460   0.0974234463
0  -2.27  44.02  0.242  810   0.2362566569
... more lines ...
22 -4.22  46.25  0.19   205   0.1181120828
21 -4.28  46.25  0.19   237   0.129990854
0  -4.73  46.25  0.19   2500  0.3346500536
5  -4.25  47.23  0.19   114   0.718192582
3  -3.72  47.25  0.19   100   0.9944669778
0  -3.25  47.25  0.19   64    1.2639918431
;
The response values are counts, so the Poisson distribution might be a reasonable model. The statistic of interest is the mackerel egg density, which can be formed as

\[
\text{Density} = \frac{\text{Eggs}}{\text{Net Area}}
\]

This is equivalent to a Poisson regression that uses the response variable \(\text{Eggs}\), an offset variable \(\log(\text{Net Area})\), and other covariates.

The following statements produce the plot of the mackerel egg density with respect to the sampling station location:

data Temp;
   set Mackerel;
   Density = Egg_Count/Net_Area;
run;

%let off0 = offsetmin=0 offsetmax=0
   linearopts=(thresholdmin=0 thresholdmax=0);
%let off0 = xaxisopts=(&off0) yaxisopts=(&off0);
proc template;
   define statgraph surface;
   dynamic _title _z;
   begingraph / designwidth=defaultDesignHeight;
   entrytitle _title;
   layout overlay / &off0;
      contourplotparm z=_z y=Latitude x=Longitude / gridded=FALSE;
   endlayout;
   endgraph;
end;
run;

proc sgrender data=Temp template=surface;
   dynamic _title='Mackerel Egg Density' _z='Density';
run;
Output 44.3.1 displays the mackerel egg density in the sampling area. The black hole in the upper right corner is caused by missing values in that area.

\textbf{Output 44.3.1} Mackerel Egg Density

In this example, the dependent variable is the mackerel egg count, the independent variables are the geographical information about each of the sampling stations, and the offset variable is the logarithm of the sampling area. The following statements use PROC GAMPL to fit the nonparametric Poisson regression model. In the program, two univariate spline terms for Depth and Distance model the nonlinear dependency of egg density on them. One bivariate spline term for both Longitude and Latitude models the nonlinear spatial effect. To allow more flexibility of the bivariate spline term, its maximum degrees of freedom is increased to 40.
data Mackerel;
  set Mackerel;
  Log_Net_Area = log(Net_Area);
run;

proc gampl data=Mackerel plots seed=321;
  model Egg_Count = spline(Depth) spline(Distance)
                  spline(Longitude Latitude/maxdf=40)
  / offset=Log_Net_Area dist=poisson;
  output out=MackerelOut1 pred=Pred1 pearson=Pea1;
run;

Output 44.3.2 displays the summary statistics from the Poisson model.

Output 44.3.2  Fit Statistics
Mackerel Egg Density Study
The GAMPL Procedure

<table>
<thead>
<tr>
<th>Fit Statistics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Penalized Log Likelihood</td>
<td>-2704.30317</td>
</tr>
<tr>
<td>Roughness Penalty</td>
<td>14.46015</td>
</tr>
<tr>
<td>Effective Degrees of Freedom</td>
<td>54.18382</td>
</tr>
<tr>
<td>Effective Degrees of Freedom for Error</td>
<td>578.23246</td>
</tr>
<tr>
<td>AIC (smaller is better)</td>
<td>5502.51384</td>
</tr>
<tr>
<td>AICC (smaller is better)</td>
<td>5512.84551</td>
</tr>
<tr>
<td>BIC (smaller is better)</td>
<td>5743.74286</td>
</tr>
<tr>
<td>UBRE (smaller is better)</td>
<td>6.58983</td>
</tr>
</tbody>
</table>

The “Tests for Smoothing Components” table in Output 44.3.3 shows that all spline terms are significant.

Output 44.3.3  Tests for Smoothing Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Effective DF</th>
<th>Effective DF for Test</th>
<th>Chi-Square</th>
<th>Pr &gt; ChiSq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline(Depth)</td>
<td>8.00000</td>
<td>8</td>
<td>243.0428</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Spline(Distance)</td>
<td>7.95797</td>
<td>9</td>
<td>171.1773</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Spline(Longitude Latitude)</td>
<td>37.22586</td>
<td>39</td>
<td>4466.2865</td>
<td>&lt;.0001</td>
</tr>
</tbody>
</table>

The smoothing component panel in Output 44.3.4 contains a contour for the spatial effect by Longitude and Latitude and two fitted univariate splines for Depth and Latitude, all demonstrating significantly nonlinear structures.
Overdispersion sometimes occurs for count data. One way to measure whether the data are overdispersed is to determine whether Pearson’s chi-square divided by the error degrees of freedom is much larger than 1. You can compute that statistic by taking the sum of squares of the Pearson residuals in the MackerelOut1 data set and then dividing that sum by the error degrees of freedom that is reported in Output 44.3.2. The computed value is approximately 8, which is much larger than 1, indicating the existence of overdispersion in this data set.

There are many ways to solve the overdispersion problem. One approach is to fit a negative binomial distribution in which the variance of the mean contains a dispersion parameter. The following statements use PROC GAMPL to fit the nonparametric negative binomial regression model:

```plaintext
proc gampl data=Mackerel plots seed=321;
  model Egg_Count = spline(Depth) spline(Distance)
  spline(Longitude Latitude/maxdf=40)
  / offset=Log_Net_Area dist=negbin;
  output out=MackerelOut2 pred=Pred2 pearson=Pea2;
run;
```
Output 44.3.5 displays the summary statistics from the negative binomial model. The model’s effective degrees of freedom is less than that of the Poisson model, even though the dispersion parameter costs one extra degree of freedom to fit. The values of the information criteria are also much less, indicating a better fit.

### Output 44.3.5  Fit Statistics

#### Mackerel Egg Density Study

**The GAMPL Procedure**

<table>
<thead>
<tr>
<th>Fit Statistics</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Penalized Log Likelihood</td>
<td>-1559.27838</td>
</tr>
<tr>
<td>Roughness Penalty</td>
<td>17.31664</td>
</tr>
<tr>
<td>Effective Degrees of Freedom</td>
<td>43.04281</td>
</tr>
<tr>
<td>Effective Degrees of Freedom for Error</td>
<td>586.85164</td>
</tr>
<tr>
<td>AIC (smaller is better)</td>
<td>3187.32573</td>
</tr>
<tr>
<td>AICC (smaller is better)</td>
<td>3193.75239</td>
</tr>
<tr>
<td>BIC (smaller is better)</td>
<td>3378.95442</td>
</tr>
<tr>
<td>UBRE (smaller is better)</td>
<td>0.55834</td>
</tr>
</tbody>
</table>

The “Tests for Smoothing Components” table in Output 44.3.6 shows that all spline terms are significant.

### Output 44.3.6  Tests for Smoothing Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Effective DF</th>
<th>Effective DF for Test</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline(Depth)</td>
<td>8.00000</td>
<td>8</td>
<td>41.21</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Spline(Distance)</td>
<td>1.00010</td>
<td>1</td>
<td>8.33</td>
<td>0.0040</td>
</tr>
<tr>
<td>Spline(Longitude Latitude)</td>
<td>32.04271</td>
<td>37</td>
<td>1024.38</td>
<td>&lt;.0001</td>
</tr>
</tbody>
</table>

The smoothing component panel in Output 44.3.7 contains a fitted surface and curves for the three spline terms. Compared to the Poisson model, the fitted surface and curves are smoother. It is possible that the nonlinear dependency structure becomes smoother after its partial variations are accounted for by the dispersion parameter.
Pearson’s chi-square divided by the error degrees of freedom for the negative binomial model is approximately 1.5, which is close to 1. This suggests that the negative binomial model explains the overdispersion in the data.

The following DATA step and statements visualize the contours of mackerel egg density for both Poisson and negative binomial models:

```sas
data SurfaceFit;
  set Mackerel(keep=Longitude Latitude Net_Area Egg_Count);
  set MackerelOut1(keep=Pred1);
  set MackerelOut2(keep=Pred2);
  Pred1 = Pred1 / Net_Area;
  Pred2 = Pred2 / Net_Area;
run;

%let eopt = location=outside valign=top textattrs=graphlabeltext;
proc template;
  define statgraph surfaces;
    begingraph / designheight=360px;
    layout lattice / columns=2;
    layout overlay / &off0;
```
Example 44.4: Nonparametric Tweedie Regression

This example illustrates how you can use the GAMPL procedure to fit a nonparametric regression model with spline effects for a response variable that has a Tweedie distribution.

You can view this model as a generalized linear model because the Tweedie distribution belongs to the exponential family. The Tweedie distribution is particularly useful for modeling response variables that are
continuous for positive values and take the value 0 with a positive probability. For example, in the insurance industry the Tweedie distribution is often assumed for claims, which are positive for customers who have filed claims and are 0 for other customers. You can use the Tweedie distribution to model such response variables without special transformations. For more information about this distribution, see the section “Tweedie Distribution For Generalized Linear Models” on page 3268.

The GENMOD procedure fits Tweedie regression models that incorporate linear effects for the covariates as illustrated in “Example 46.12: Tweedie Regression” on page 3356. However, as demonstrated in this example, the assumption of linearity can be too restrictive, and you might want to use the GAMPL procedure to explore the nonlinear dependency structures in the data.

The following DATA step simulates a response variable $Y$ by using a Tweedie distribution and four continuous predictors. Each continuous predictor is sampled independently from the uniform distribution $U(0, 1)$. The linear predictor $\eta$ for the true model is formed by applying additive nonlinear transformations to $x_1$, $x_2$, and $x_3$:

$$\eta = \frac{2 \sin(\pi x_1) + 0.8 \exp(2x_2) + 0.2x_3^{11}(10(1-x_3))^6 + 10(10x_3)^3(1-x_3)^{10}}{20}$$

The predictor $x_4$ is a nuisance parameter that does not enter the model. The parameters for the Tweedie distribution are $\phi = 0.4$ and $p = 1.5$, and the link function is $\log \mu = \eta$, where the expected value is $E(Y) = \mu$ and the variance is $\text{Var}(Y) = \phi \mu^p$. The power parameter $p$ controls the variance of the distribution. When $p$ is between 1 and 2, as in this example, a Tweedie random variable can be generated from a compound Poisson distribution (Smyth 1996).

```plaintext
title 'Nonparametric Tweedie Model';
%let phi=0.4;
%let power=1.5;

data one;
  do i=1 to 1000;
    /* Sample the predictors */
    x1=ranuni(1);
    x2=ranuni(1);
    x3=ranuni(1);
    x4=ranuni(1);

    /* Apply nonlinear transformations to predictors */
    f1=2*sin(3.14159265*x1);
    f2=exp(2*x2)*0.8;
    f3=0.2*x3**11*(10*(1-x3))**6+10*(10*x3)**3*(1-x3)**10;
    xb=f1+f2+f3;
    xb=xb/20;
    mu=exp(xb);

    /* Compute parameters of compound Poisson distribution */
    lambda=mu**((2-&power)/(&phi*(2-&power)));
    alpha=(2-&power)/(&power-1);
    gamma=&phi*(&power-1)*(+mu**(&power-1))

    /* Simulate the response */
    rpoi=ranpoi(1,lambda);
  end;
```

if rpoi=0 then y=0;
else do;
  y=0;
  do j=1 to rpoi;
    y=y+rangam(1,alpha);
  end;
  y=y*gamma;
end;
output;
end;
run;

You can use PROC GENMOD to fit a parametric model for \( Y \) by using the following statements:

```plaintext
proc genmod data=one;
  model y=x1 x2 x3 x4/dist=tweedie;
run;
```

Equivalently, you can also use PROC GAMPL to fit a parametric linear model by using the PARAM option in the MODEL statement, as shown in the following statements.

```plaintext
proc gampl data=one seed=1234;
  model y=param(x1 x2 x3 x4)/dist=tweedie;
run;
```

The “Fit Statistics” table in Output 44.4.1 shows the summary statistics for the fitted parametric Tweedie model.

### Output 44.4.1  Fit Statistics

#### Nonparametric Tweedie Model

The GAMPL Procedure

<table>
<thead>
<tr>
<th>Fit Statistics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Penalized Log Likelihood</td>
<td>-1178.69507</td>
</tr>
<tr>
<td>Roughness Penalty</td>
<td>0</td>
</tr>
<tr>
<td>Effective Degrees of Freedom</td>
<td>7.00000</td>
</tr>
<tr>
<td>Effective Degrees of Freedom for Error</td>
<td>993.00000</td>
</tr>
<tr>
<td>AIC (smaller is better)</td>
<td>2371.39014</td>
</tr>
<tr>
<td>AICC (smaller is better)</td>
<td>2371.50305</td>
</tr>
<tr>
<td>BIC (smaller is better)</td>
<td>2405.74443</td>
</tr>
</tbody>
</table>

The “Parameter Estimates” table in Output 44.4.2 shows the estimates for regression parameters in addition to dispersion and power parameters (\( \phi \) and \( p \)).
If you are uncertain about the relationship between the linked mean for the fitted Tweedie model and the covariates, you can add spline terms to the model to explore possible nonlinear dependence. The following statements fit a nonparametric Tweedie model and produce a plot of fitted smoothing components.

```plaintext
proc gampl data=one seed=1234 plots;
   model y=spline(x1) spline(x2) spline(x3) spline(x4)/dist=tweedie;
run;
```

The “Fit Statistics” table in Output 44.4.3 shows the summary statistics for the fitted nonparametric Tweedie model. Compared to the parametric model, the nonparametric model has more effective degrees of freedom, but smaller AIC and AICC values. The penalized likelihood value is larger for the nonparametric model. These statistics indicate that the nonparametric model provides a better fit.

```
Output 44.4.3  Fit Statistics

Nonparametric Tweedie Model

The GAMPL Procedure

<table>
<thead>
<tr>
<th>Fit Statistics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Penalized Log Likelihood</td>
<td>-1157.17502</td>
</tr>
<tr>
<td>Roughness Penalty</td>
<td>2.62230</td>
</tr>
<tr>
<td>Effective Degrees of Freedom</td>
<td>14.11920</td>
</tr>
<tr>
<td>Effective Degrees of Freedom for Error</td>
<td>983.62098</td>
</tr>
<tr>
<td>AIC (smaller is better)</td>
<td>2339.96615</td>
</tr>
<tr>
<td>AICC (smaller is better)</td>
<td>2340.39965</td>
</tr>
<tr>
<td>BIC (smaller is better)</td>
<td>2409.25974</td>
</tr>
<tr>
<td>GCV (smaller is better)</td>
<td>0.39845</td>
</tr>
</tbody>
</table>
```

The “Regression Parameter Estimates” table in Output 44.4.4 lists the estimate for the intercept, in addition to the estimates for the dispersion and power parameters for the Tweedie model. Both variance parameters are close to the true values.
Output 44.4 Regression Parameter Estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>DF</th>
<th>Estimate</th>
<th>Standard Error</th>
<th>Chi-Square</th>
<th>Pr &gt; ChiSq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>1</td>
<td>0.356825</td>
<td>0.018329</td>
<td>379.0079</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Dispersion</td>
<td>1</td>
<td>0.400459</td>
<td>0.518084</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>1</td>
<td>1.501771</td>
<td>1.514449</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The “Estimates for Smoothing Components” table in Output 44.4.5 shows the fitted information for the spline terms in the model. The degrees of freedom value and the roughness penalty value suggest some moderate nonlinear relationship between the linked mean and the three covariates $x_1$, $x_2$, and $x_3$. The spline term for the nuisance parameter $x_4$ has a linear form.

Output 44.4.5 Estimates for Smoothing Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Effective DF</th>
<th>Smoothing Parameter</th>
<th>Roughness Penalty</th>
<th>Number of Parameters</th>
<th>Rank of Penalty Matrix</th>
<th>Number of Knots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline(x1)</td>
<td>2.10969</td>
<td>0.9560</td>
<td>0.5659</td>
<td>9</td>
<td>10</td>
<td>1000</td>
</tr>
<tr>
<td>Spline(x2)</td>
<td>2.38340</td>
<td>0.5604</td>
<td>0.3814</td>
<td>9</td>
<td>10</td>
<td>1000</td>
</tr>
<tr>
<td>Spline(x3)</td>
<td>5.62608</td>
<td>0.0145</td>
<td>1.6750</td>
<td>9</td>
<td>10</td>
<td>1000</td>
</tr>
<tr>
<td>Spline(x4)</td>
<td>1.0002</td>
<td>133078</td>
<td>3.662E-6</td>
<td>9</td>
<td>10</td>
<td>1000</td>
</tr>
</tbody>
</table>

The “Tests for Smoothing Components” table in Output 44.4.6 shows the approximate Wald test for the four spline terms. The spline term for the nuisance predictor $x_4$ is not significant.

Output 44.4.6 Tests for Smoothing Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Effective DF</th>
<th>Effective DF for Test</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline(x1)</td>
<td>2.10969</td>
<td>3</td>
<td>4.76</td>
<td>0.0027</td>
</tr>
<tr>
<td>Spline(x2)</td>
<td>2.38340</td>
<td>3</td>
<td>13.64</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Spline(x3)</td>
<td>5.62608</td>
<td>7</td>
<td>49.02</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Spline(x4)</td>
<td>1.0002</td>
<td>1</td>
<td>0.88</td>
<td>0.3484</td>
</tr>
</tbody>
</table>

Output 44.4.7 displays the “Smoothing Component Panel” for all the spline terms used in the model. For $x_1$, $x_2$, and $x_3$, the fitted curves are reasonably smooth and close to the true functions. For $x_4$, the spline plot shows a strict linear fit, with 95% Bayesian confidence bands covering the horizontal line at 0. This reinforces the conclusion that $x_4$ does not contribute to the model and can be removed from further analysis.
Output 44.4.7  Smoothing Component Panel

Smoothing Components for y

With 95% Confidence Bands

DF = 2.11

0.05
0.00
-0.05
-0.10
-0.15

0.2
0.4
0.6
0.8

Spline(x1)

DF = 2.383

0.2
0.4
0.6
0.8

Spline(x2)

DF = 5.626

0.2
0.4
0.6
0.8

Spline(x3)

DF = 1

0.2
0.4
0.6
0.8

Spline(x4)
References


Subject Index

class level
  GAMPL procedure, 3041, 3072

computational method
  GAMPL procedure, 3067

convergence criterion
  GAMPL procedure, 3040, 3043, 3044

convergence status
  GAMPL procedure, 3073

data access information
  GAMPL procedure, 3072

degrees of freedom
  GAMPL procedure, 3064

dispersion parameter
  GAMPL procedure, 3065

displayed output
  GAMPL procedure, 3071

distribution function
  GAMPL procedure, 3052

effect
  name length (GAMPL), 3040

estimates for smoothing components
  GAMPL procedure, 3074

fit statistics
  GAMPL procedure, 3073

fitting algorithms
  GAMPL procedure, 3063

frequency variable
  GAMPL procedure, 3046

GAMPL procedure, 3029
  class level, 3041, 3072
  computational method, 3067
  convergence criterion, 3040, 3043, 3044
  convergence status, 3073
  data access information, 3072
  degrees of freedom, 3064
  dispersion parameter, 3065
  displayed output, 3071
  distribution function, 3052
  effect name length, 3040
  estimates for smoothing components, 3074
  fit statistics, 3073
  fitting algorithms, 3063
  function-based convergence criteria, 3043
  generalized additive models, 3060
  generalized linear models, 3060
  gradient-based convergence criteria, 3043, 3044
  input data sets, 3040
  iteration history, 3040
  link function, 3054
  low-rank approximation, 3059
  model evaluation criteria, 3062
  model inference, 3065
  model information, 3072
  model options summary, 3047
  multithreading, 3057, 3067
  number of observations, 3072
  ODS Graphics, 3075
  ODS table names, 3074
  optimization technique, 3068
  outer iteration, 3063
  parameter estimates, 3074
  penalized likelihood estimation, 3061
  performance information, 3071
  performance iteration, 3064
  response level ordering, 3048
  response profile, 3072
  response variable options, 3048
  spline details, 3072
  tests for smoothing components, 3066, 3074
  thin-plate regression splines, 3057
  thin-plate smoothing splines, 3058
  user-defined formats, 3040
  weighting, 3057
  XML input stream, 3040

generalized additive models
  GAMPL procedure, 3060

generalized linear models
  GAMPL procedure, 3060

iteration history
  GAMPL procedure, 3073

link function
  GAMPL procedure, 3054

low-rank approximation
  GAMPL procedure, 3059

model evaluation criteria
  GAMPL procedure, 3062

model inference
  GAMPL procedure, 3065

model information
  GAMPL procedure, 3072

multithreading
number of observations
   GAMPL procedure, 3072

ODS Graphics
   GAMPL procedure, 3075
ODS table names
   GAMPL procedure, 3074
optimization technique
   GAMPL procedure, 3068
options summary
   PROC GAMPL statement, 3039
outer iteration
   GAMPL procedure, 3063

parameter estimates
   GAMPL procedure, 3074
penalized likelihood estimation
   GAMPL procedure, 3061
performance information
   GAMPL procedure, 3071
performance iteration
   GAMPL procedure, 3064

response level ordering
   GAMPL procedure, 3048
response profile
   GAMPL procedure, 3072
response variable options
   GAMPL procedure, 3048
reverse response level ordering
   GAMPL procedure, 3048

spline details
   GAMPL procedure, 3072

tests for smoothing components
   GAMPL procedure, 3066, 3074
thin-plate regression splines
   GAMPL procedure, 3057
thin-plate smoothing splines
   GAMPL procedure, 3058

weighting
   GAMPL procedure, 3057
Syntax Index

ABSCONV option
  PROC GAMPL statement, 3043
ABSFCONV option
  PROC GAMPL statement, 3043
ABSGCONV option
  PROC GAMPL statement, 3043
ABSGTOL option
  PROC GAMPL statement, 3043
ABSTOL option
  PROC GAMPL statement, 3043
ALLOBS option
  MODEL statement (GAMPL), 3052
ALPHA= option
  OUTPUT statement (GAMPL), 3056
  PROC GAMPL statement, 3040
CLASS statement
  GAMPL procedure, 3045
COMPONENT option
  OUTPUT statement (GAMPL), 3056
  PROC GAMPL statement, 3040
CRITERION= option
  MODEL statement (GAMPL), 3052
DATA= option
  OUTPUT statement (GAMPL), 3056
  PROC GAMPL statement, 3040
DESCENDING option
  MODEL statement (GAMPL), 3048
DETAILS option
  MODEL statement (GAMPL), 3050
DF= option
  MODEL statement (GAMPL), 3050
DISPERSION= option
  MODEL statement (GAMPL), 3052
DISTRIBUTION= option
  MODEL statement (GAMPL), 3052
FCONV option
  PROC GAMPL statement, 3043
FDHESSIAN= option
  MODEL statement (GAMPL), 3053
FMTLIBXML= option
  PROC GAMPL statement, 3040
FREQ statement
  GAMPL procedure, 3046
FTOL option
  PROC GAMPL statement, 3043
GAMPL procedure, 3039
CLASS statement, 3045
FREQ statement, 3046
ID statement, 3046
MODEL statement, 3046
OUTPUT statement, 3055
PERFORMANCE statement, 3057
PROC GAMPL statement, 3039
syntax, 3039
WEIGHT statement, 3057
GAMPL procedure, CLASS statement, 3045
GAMPL procedure, FREQ statement, 3046
GAMPL procedure, ID statement, 3046
GAMPL procedure, MODEL statement, 3046
  ALLOBS option, 3052
  CRITERION= option, 3052
  DESCENDING option, 3048
  DETAILS option, 3050
  DF= option, 3050
  DISPERSION= option, 3052
  DISTRIBUTION= option, 3052
  FDHESSIAN option, 3053
  INITIALPHI= option, 3054
  INITSMOOTH= option, 3050
  KNITS= option, 3050
  LINK= option, 3054
  M= option, 3051
  MAXDF= option, 3051
  MAXKNOTS= option, 3051
  MAXPHI= option, 3054
  MAXSMOOTH= option, 3051
  METHOD= option, 3054
  MINPHI= option, 3054
  MINSMOOTH= option, 3052
  NORMALIZE option, 3054
  OFFSET= option, 3055
  ORDER= option, 3049
  RIDGE= option, 3055
  SCALE= option, 3055
  SMOOTH= option, 3052
GAMPL procedure, OUTPUT statement, 3055
  ALPHA= option, 3056
  COMPONENT option, 3056
  DATA= option, 3056
  keyword= option, 3056
  OUT= option, 3056
GAMPL procedure, PERFORMANCE statement, 3057
GAMPL procedure, PROC GAMPL statement, 3039
ABSCONV option, 3043
ABSFCONV option, 3043
ABSFTOL option, 3043
ABSGCONV option, 3043
ABSGTOL option, 3043
ABSTOL option, 3043
ALPHA= option, 3040
DATA= option, 3040
FCONV option, 3043
FMTLIBXML= option, 3040
FTOL option, 3043
GCONV option, 3044
GTOL option, 3044
ITDETAILS option, 3040
MAXFUNC= option, 3044
MAXITER= option, 3044, 3045
MAXKNOTS= option
MODEL statement (GAMPL), 3051
MAXPHI= option
MODEL statement (GAMPL), 3054
MAXSMOOTH= option
MODEL statement (GAMPL), 3051
MAXTIME= option
PROC GAMPL statement, 3045
METHOD= option
MODEL statement (GAMPL), 3054
MINPHI= option
MODEL statement (GAMPL), 3054
MINSMOOTH= option
MODEL statement (GAMPL), 3052
MODEL statement
GAMPL procedure, 3046
NAMELEN= option
PROC GAMPL statement, 3040
NOCLPRINT option
PROC GAMPL statement, 3041
NOPRINT option
PROC GAMPL statement, 3041
NORMALIZE option
MODEL statement (GAMPL), 3054
OFFSET= option
MODEL statement (GAMPL), 3055
ORDER= option
MODEL statement (GAMPL), 3049
OUT= option
OUTPUT statement (GAMPL), 3056
OUTPUT statement
GAMPL procedure, 3055
PERFORMANCE statement
GAMPL procedure, 3057
PLIKEOPTIONS option
PROC GAMPL statement, 3041
PLOTS= option
PROC GAMPL statement, 3041
PROC GAMPL statement, see GAMPL procedure
RIDGE= option
MODEL statement (GAMPL), 3055
SCALE= option
MODEL statement (GAMPL), 3055
SEED= option
PROC GAMPL statement, 3042
GAMPL procedure, WEIGHT statement, 3057
GCONV option
PROC GAMPL statement, 3044
GTOL option
PROC GAMPL statement, 3044
ID statement
GAMPL procedure, 3046
INITIALPHI= option
MODEL statement (GAMPL), 3054
INITSMOOTH= option
MODEL statement (GAMPL), 3050
ITDETAILS option
PROC GAMPL statement, 3040
keyword= option
OUTPUT statement (GAMPL), 3056
KNOTS= option
MODEL statement (GAMPL), 3050
LINK= option
MODEL statement (GAMPL), 3054
M= option
MODEL statement (GAMPL), 3051
MAXDF= option
PROC GAMPL statement, 3044
MODEL statement (GAMPL), 3051
PROC GAMPL statement, 3044
PROC GAMPL statement, 3044, 3045
PROC GAMPL statement (GAMPL), 3051
PROC GAMPL statement (GAMPL), 3054
PROC GAMPL statement (GAMPL), 3051
PROC GAMPL statement (GAMPL), 3052
PROC GAMPL statement
GAMPL procedure, 3046
SINGCHOL= option
   PROC GAMPL statement, 3042
SINGSWEEP= option
   PROC GAMPL statement, 3042
SINGULAR= option
   PROC GAMPL statement, 3042
SMOOTH= option
   MODEL statement (GAMPL), 3052
SMOOTHOPTIONS option
   PROC GAMPL statement, 3042
syntax
   GAMPL procedure, 3039

TECHNIQUE= option
   PROC GAMPL statement, 3045

unpack option
   PROC GAMPL statement, 3041

WEIGHT statement
   GAMPL procedure, 3057