Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore for additional books and resources.
Overview: HPMIXED Procedure

The HPMIXED procedure uses a number of specialized high-performance techniques to fit linear mixed models with variance component structure. The HPMIXED procedure is specifically designed to cope with estimation problems involving a large number of fixed effects, a large number of random effects, or a large number of observations.

The HPMIXED procedure complements the MIXED procedure and other SAS/STAT procedures for mixed modeling. On the one hand, the models supported by the HPMIXED procedure are a subset of the models that you can fit with the MIXED procedure, and the confirmatory inferences available in the HPMIXED procedure are also a subset of the general analyses available with the MIXED procedure. On the other hand, the HPMIXED procedure can have considerably better performance than other SAS/STAT mixed modeling tools, in terms of memory requirements and computational speed.

A mixed model can be large in a number of ways, not all of which are suited for the specialized algorithms and storage techniques implemented in the HPMIXED procedure. The following are examples of linear mixed modeling problems for which the HPMIXED procedure has been specifically designed:

- linear mixed models with thousands of levels for the fixed and/or random effects
- linear mixed models with hierarchically nested fixed and/or random effects, possibly with hundreds or thousands of levels at each level of the hierarchy

Basic Features

The HPMIXED procedure enables you to specify a linear mixed model with variance component structure, to estimate the covariance parameters by restricted maximum likelihood, and to perform confirmatory inference in such models. The HPMIXED procedure fits the specified linear mixed model and produces appropriate statistics.

The following are some of the basic features of the HPMIXED procedure:

- capacity to handle large linear mixed model problems for balanced or unbalanced data
- MIXED-type MODEL and RANDOM statements for model specification and CONTRAST, ESTIMATE, LSMEANS, and TEST statements for inferences
- estimate covariance parameters by restricted maximum likelihood (REML)
- output statistics by using the OUTPUT statement
• computation of appropriate standard errors for all specified estimable linear combinations of fixed and random effects, and corresponding $t$ and $F$ tests

• subject and group effects that enable blocking and heterogeneity, respectively

• NLOPTIONS statement, which enables you to exercise control over the numerical optimization

The HPMIXED procedure uses the Output Delivery System (ODS), a SAS subsystem that provides capabilities for displaying and controlling the output from SAS procedures. ODS enables you to convert any of the output from the HPMIXED procedure into a SAS data set. See the section “ODS Table Names” on page 3863 and Chapter 20, “Using the Output Delivery System,” for further information about using ODS with the HPMIXED procedure.

Assumptions and Notation

The linear mixed models fit by the HPMIXED procedure can be represented as linear statistical models in the following form:

$$y = X\beta + Z\gamma + \epsilon$$

$$\gamma \sim N(0, G)$$

$$\epsilon \sim N(0, \sigma^2 I)$$

$$\text{Cov}[y, \epsilon] = 0$$

The symbols in these expressions denote the following:

$y$ the $(n \times 1)$ vector of responses

$X$ the $(n \times k)$ design matrix for the fixed effects

$\beta$ the $(k \times 1)$ vector of fixed-effects parameters

$Z$ the $(n \times q)$ design matrix for the random effects

$\gamma$ the $(q \times 1)$ vector of random effects

$\epsilon$ the $(n \times 1)$ vector of unobservable residual errors

As is customary for statistical models in the linear mixed model family, the random effects are assumed normally distributed. The same holds for the residual errors and these are furthermore distributed independently of the random effects. As a consequence, these assumptions imply that the response vector $y$ has a multivariate normal distribution.

Further assumptions, implicit in the preceding expression, are as follows:

• The conditional mean of the data—given the random effects—is linear in the fixed effects and the random effects.

• The marginal mean of the data is linear in the fixed-effects parameters.
Computational Approach

The computational methods to efficiently solve large mixed model problems with the HPMIXED procedure rely on a combination of several techniques, including sparse matrix storage, specialized solving of sparse linear systems, and dedicated nonlinear optimization.

Sparse Storage and Computation

One of the fundamental computational tasks in analyzing a linear mixed model is solving the mixed model equations

\[
\begin{bmatrix}
X'X & X'Z \\
Z'X & Z'Z + \sigma^2 G^{-1}
\end{bmatrix}
\begin{bmatrix}
\beta \\
y
\end{bmatrix} =
\begin{bmatrix}
X'y \\
y'
\end{bmatrix}
\]

where \( G \) denotes the variance matrix of the random effects. The mixed model crossproduct matrix

\[
\begin{bmatrix}
X'X & X'Z \\
Z'X & Z'Z + \sigma^2 G^{-1}
\end{bmatrix}
\]

is a key component of these equations, and it often has many zero values (George and Liu 1981). Sparse storage techniques can result in significant savings in both memory and CPU resources. The HPMIXED procedure draws on sparse matrix representation and storage where appropriate or necessary.

Conjugate Gradient Algorithm and Iteration-on-Data Technology

Solving the mixed model equations is a critical component of linear mixed model analysis. The two main components of the preconditioned conjugate gradient (PCCG) algorithm are preconditioning and matrix-vector product computing (Shewchuk 1994). The algorithm is guaranteed to converge to the solution within \( n_e \) iterations, where \( n_e \) is equal to the number of distinct eigenvalues of the mixed model equations. This simple yet powerful algorithm can be easily implemented with an iteration-on-data (IOD) technique (Tsuruta, Misztal, and Stranden 2001) that can yield significant savings of memory resources.

The combination of the PCCG algorithm and iteration on data makes it possible to efficiently compute best linear unbiased predictors (BLUPs) for the random effects in mixed models with large mixed model equations.

Average Information Algorithm

The HPMIXED procedure estimates covariance parameters by restricted maximum likelihood. The default optimization method is a quasi-Newton algorithm. When the Hessian or information matrix is required, the HPMIXED procedure takes advantage of the computational simplifications that are available by averaging information (AI). The AI algorithm (Johnson and Thompson 1995; Gilmour, Thompson, and Cullis 1995) replaces the second derivative matrix with the average of the observed and expected information matrices. The computationally intensive trace terms in these information matrices cancel upon averaging. Coarsely, the AI algorithm can be viewed as a hybrid of a Newton-Raphson approach and Fisher scoring.
The HPMIXED Procedure Contrasted with the MIXED Procedure

The HPMIXED procedure is designed to solve large mixed model problems by using sparse matrix techniques. A mixed model can be large in many ways: a large number of observations, a large number of columns in the $X$ matrix, a large number of columns in the $Z$ matrix, and a large number of covariance parameters. The aim of the HPMIXED procedure is parameter estimation, inference, and prediction in linear mixed models with large $X$ and/or $Z$ matrices and many observations, but with relatively few covariance parameters.

The models that you can fit with the HPMIXED procedure and the available postprocessing analyses are a subset of the models and analyses available with the MIXED procedure. With the HPMIXED procedure you can model only G-side random effects with variance component structure or an unstructured covariance matrix in a Cholesky parameterization. R-side random effects and direct modeling of their covariance structures are not supported.

The MIXED and HPMIXED procedures offer different balances for computing performance and statistical generality. To some extent the generality of the MIXED procedure means that it cannot serve as a high-performance computing tool for all of the model-data scenarios that it can potentially handle. For example, although efficient sparse algorithms are available to estimate variance components in large linear mixed models, the computational configuration changes profoundly when, for example, Kenward-Roger degree-of-freedom adjustments are requested.

On the other hand, the HPMIXED procedure can handle only a small subset of the models that PROC MIXED can fit. Invariably, some features of high-performance sparse computing methods might be surprising at first. For example, the best computational path depends on the model and the data, so that in models with a singular $X'X$ matrix, the order in which singularities are detected and accounted for can change from one data set to the next.

The following is a list of features available in the MIXED procedure, but not available in the HPMIXED procedure:

- a variety of covariance structures by using the TYPE= option in the RANDOM statement
- automatic Type III tests of fixed effects. You request tests of fixed effects in the HPMIXED procedure with the TEST statement.
- ODS statistical graphics
- advanced degree-of-freedom adjustments available by using the DDFM= option
- maximum likelihood or method-of-moments estimation for the covariance parameters
- a PRIOR statement for a sampling-based Bayesian analysis
In animal breeding, it is common to model genetic and environmental effects with a random effect for the animal. When there are many animals being studied, this can lead to very large mixed model equations to be solved. In this example we present an analysis of simulated data with this structure.

Suppose you have 3000 animals from five different genetic species raised on 100 different farms. The following DATA step simulates 40000 observations of milk yield (Yield) from a linear mixed model with variables Species and Farm in the fixed-effect model and Animal as a random effect. The random effect due to Animal is simulated with a variance of 4.0, while the residual error variance is 8.0. These variance component values reflect the fact that variation in milk yield is typically genetically controlled to be no more than 33% (4/(4+8)).

```sas
data Sim;
  keep Species Farm Animal Yield;
  array AnimalEffect{3000};
  array AnimalFarm{3000};
  array AnimalSpecies{3000};
  do i = 1 to dim(AnimalEffect);
    AnimalEffect{i} = sqrt(4.0) * rannor(12345);
    AnimalFarm{i} = 1 + int(100 * ranuni(12345));
    AnimalSpecies{i} = 1 + int(5 * ranuni(12345));
  end;
  do i = 1 to 40000;
    Animal = 1 + int(3000 * ranuni(12345));
    Species = AnimalSpecies{Animal};
    Farm = AnimalFarm{Animal};
    Yield = 1 + Species + Farm/10 + AnimalEffect{Animal} + sqrt(8.0) * rannor(12345);
    output;
  end;
run;
```

A simple linear mixed model analysis is performed by using the following SAS statements:

```sas
proc hpmixed data=Sim;
  class Species Farm Animal;
  model Yield = Species Species*Farm;
  random Animal;
  test Species*Farm;
  contrast 'Species1 = Species2 = Species3'
    Species 1 0 -1,
    Species 0 1 -1;
run;
```
Selected results from the preceding SAS statements are shown in Figure 49.1 through Figure 49.4.

The “Class Level Information” table in Figure 49.1 shows that the three model effects have 5, 100, and 3000 levels, respectively. Only a portion of the levels are displayed by default. The “Dimensions” table shows that the model contains a single G-side covariance parameter and a single R-side covariance parameter. R-side covariance parameters are those associated with the covariance matrix $R$ in the conditional distribution, given the random effects. In the case of the HPMIXED procedure this matrix is simply $R = \sigma^2 I$ and the single R-side covariance parameter corresponds to the residual variance. The G-side parameter is the variance of the random Animal effect; the $G$ matrix is a diagonal $(3000 \times 3000)$ matrix with the common variance on the diagonal.

![Figure 49.1](image.png)

**Figure 49.1** Class Levels and Dimensions

<table>
<thead>
<tr>
<th>Class</th>
<th>Levels</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Species</td>
<td>5 1 2 3 4 5</td>
<td></td>
</tr>
<tr>
<td>Farm</td>
<td>100 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ...</td>
<td></td>
</tr>
<tr>
<td>Animal</td>
<td>3000 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ...</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-side Cov. Parameters</td>
</tr>
<tr>
<td>R-side Cov. Parameters</td>
</tr>
<tr>
<td>Columns in $X$</td>
</tr>
<tr>
<td>Columns in $Z$</td>
</tr>
<tr>
<td>Subjects (Blocks in $V$)</td>
</tr>
</tbody>
</table>

Taking into account the intercept as well as the number of levels of the Species and Species*Farm effects, the $X$ matrix for this problem has 506 columns, so that the mixed model equations

$$
\begin{bmatrix}
X'X & X'Z \\
Z'X & Z'Z + \sigma^2 G^{-1}
\end{bmatrix}
\begin{bmatrix}
\beta \\
\gamma
\end{bmatrix}
= 
\begin{bmatrix}
X'y \\
Z'y
\end{bmatrix}
$$

have 3506 rows and columns. This is a substantial computational problem: simply storing a single copy of this matrix in dense format requires nearly 50 megabytes of memory. The sparse matrix techniques of PROC HPMIXED use a small fraction of this amount of memory and a similarly small fraction of the CPU time required to solve the equations with dense techniques. For more information about sparse versus dense techniques, see the section “Sparse Matrix Techniques” on page 3860.
Figure 49.2 displays the covariance parameter estimates at convergence of the REML algorithm. The variance component estimate for animal effect is $\hat{\sigma}_a^2 = 3.9889$ and for residual $\hat{\sigma}_e^2 = 7.9623$. These estimates are close to the simulated values (4.0 and 8.0).

**Figure 49.2  Estimates of Variance Components**

<table>
<thead>
<tr>
<th>Cov Parm</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animal</td>
<td>3.9889</td>
</tr>
<tr>
<td>Residual</td>
<td>7.9623</td>
</tr>
</tbody>
</table>

The TEST statement requests a Type III test of the fixed effect in the model. By default, the HPMIXED procedure does not compute Type III tests, because they can be computationally demanding. The tests of the Species*Farm effect is highly significant. That indicates animals of a genetic species perform differently in different environments.

**Figure 49.3  Type III Tests of Fixed Effect**

<table>
<thead>
<tr>
<th>Effect</th>
<th>Num DF</th>
<th>Den DF</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Species*Farm</td>
<td>495</td>
<td>39500</td>
<td>11.72</td>
<td>&lt;.0001</td>
</tr>
</tbody>
</table>

You can use the CONTRAST or ESTIMATE statement to test custom linear hypotheses involving the fixed and/or random effects. The CONTRAST statement in the preceding program tests the null hypothesis that there are no differences among the first three genetic species. Results from this analysis are shown in Figure 49.4. The small $p$-value indicates that there are significant differences among the first three genetics species.

**Figure 49.4  Result of CONTRAST Statement**

<table>
<thead>
<tr>
<th>Contrasts</th>
<th>Num DF</th>
<th>Den DF</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Species1 = Species2 = Species3</td>
<td>2</td>
<td>39500</td>
<td>92.93</td>
<td>&lt;.0001</td>
</tr>
</tbody>
</table>
Syntax: HPMIXED Procedure

The following statements are available in the HPMIXED procedure:

\[
\text{PROC HPMIXED} \ < \text{options}>; \\
\quad \text{BY} \ \text{variables}; \\
\quad \text{CLASS} \ \text{variable} \ < \text{(REF= option)} > \ldots \ < \text{variable} \ < \text{(REF= option)}> \ < \text{global-options}>; \\
\quad \text{EFFECT} \ \text{name}=\text{effect-type} \ \text{(variables} < \text{options}) ; \\
\quad \text{ID} \ \text{variables}; \\
\quad \text{MODEL} \ \text{dependent} = < \text{fixed-effects}> < \text{options}>; \\
\quad \text{RANDOM} \ \text{random-effects} < \text{options}>; \\
\quad \text{REPEATED} \ \text{repeated-effect} < \text{options}>; \\
\quad \text{PARMS} < \text{value-list} > < \text{options}>; \\
\quad \text{TEST} \ \text{fixed-effects} < \text{options}>; \\
\quad \text{CONTRAST} \ \text{'label'} \ \text{contrast-specification} <, \ \text{contrast-specification} > <, \ldots > < \text{options}>; \\
\quad \text{ESTIMATE} \ \text{'label'} \ \text{contrast-specification} < (\text{divisor=} n) > \\
\quad \text{LSMEANS} \ \text{fixed-effects} < \text{options}>; \\
\quad \text{NLOPTIONS} < \text{options}>; \\
\quad \text{OUTPUT} < \text{OUT=SAS-data-set}> \\
\quad \text{WEIGHT} \ \text{variable};
\]

Items within angle brackets ( < > ) are optional. The CONTRAST, ESTIMATE, LSMEANS, RANDOM, and TEST statements can appear multiple times; all other statements can appear only once.

The PROC HPMIXED and MODEL statements are required, and the MODEL statement must appear after the CLASS statement if these statements are included. The BY, CLASS, MODEL, ID, OUTPUT, TEST, RANDOM, REPEATED and WEIGHT statements are described in full after the PROC HPMIXED statement in alphabetical order. The EFFECT, is shared with many other procedures. Summary descriptions of functionality and syntax for this statement is also given after the PROC HPMIXED statement in alphabetical order, but you can find full documentation on it in Chapter 19, “Shared Concepts and Topics.”

Table 49.1 summarizes the basic functions and important options of each PROC HPMIXED statement.

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROC HPMIXED</td>
<td>Invokes the procedure</td>
<td>DATA= specifies input data set, METHOD= specifies estimation method</td>
</tr>
<tr>
<td>BY</td>
<td>Performs multiple PROC HPMIXED analyses in one invocation</td>
<td>None</td>
</tr>
<tr>
<td>CLASS</td>
<td>Declares qualitative variables that create indicator variables in design matrices</td>
<td>None</td>
</tr>
</tbody>
</table>
Table 49.1 continued

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Lists additional variables to be included in predicted values tables</td>
<td>None</td>
</tr>
<tr>
<td>MODEL</td>
<td>Specifies dependent variable and fixed effects, setting up X</td>
<td>S requests solution for fixed-effects parameters, DDFM= specifies denominator degrees of freedom method</td>
</tr>
<tr>
<td>RANDOM</td>
<td>Specifies random effects, setting up Z and G</td>
<td>SUBJECT= creates block-diagonality, TYPE= specifies covariance structure, S requests solution for random-effects parameters</td>
</tr>
<tr>
<td>REPEATED</td>
<td>Sets up R</td>
<td>SUBJECT= creates block-diagonality, TYPE= specifies covariance structure, R= displays estimated blocks of R, GROUP= enables between-subject heterogeneity</td>
</tr>
<tr>
<td>PARMS</td>
<td>Specifies a grid of initial values for the covariance parameters</td>
<td>HOLD= and NOITER hold the covariance parameters or their ratios constant, PARMS- DATA= reads the initial values from a SAS data set</td>
</tr>
<tr>
<td>CONTRAST</td>
<td>Constructs custom hypothesis tests</td>
<td>E displays the L matrix coefficients</td>
</tr>
<tr>
<td>ESTIMATE</td>
<td>Constructs custom scalar estimates</td>
<td>CL produces confidence limits</td>
</tr>
<tr>
<td>LSMEANS</td>
<td>Computes least squares means for classification fixed effects</td>
<td>DIFF computes differences of the least squares means, CL produces confidence limits, SLICE= tests simple effects</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>Specifies a variable by which to weight R</td>
<td>None</td>
</tr>
</tbody>
</table>

PROC HPMIXED Statement

PROC HPMIXED <options> ;

The PROC HPMIXED statement invokes the HPMIXED procedure. Table 49.2 summarizes the options available in the PROC HPMIXED statement. These and other options in the PROC HPMIXED statement are then described fully in alphabetical order.

Table 49.2 PROC HPMIXED Statement Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA=</td>
<td>Specifies input data set</td>
</tr>
<tr>
<td>METHOD=</td>
<td>Specifies the estimation method</td>
</tr>
<tr>
<td>NOPROFILE</td>
<td>Includes scale parameter in optimization</td>
</tr>
</tbody>
</table>
### Table 49.2 continued

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORDER=</td>
<td>Determines the sort order of CLASS variables</td>
</tr>
<tr>
<td>BLUP</td>
<td>Computes BLUP/BLUE only</td>
</tr>
<tr>
<td>IC=</td>
<td>Displays a table of information criteria</td>
</tr>
<tr>
<td>ITDETAILS</td>
<td>Displays estimates and gradients added to “Iteration History”</td>
</tr>
<tr>
<td>MAXCLPRINT=</td>
<td>Specifies the maximum levels of CLASS variables to print</td>
</tr>
<tr>
<td>LOGNOTE</td>
<td>Writes periodic status notes to the log</td>
</tr>
<tr>
<td>MMEQ</td>
<td>Displays mixed model equations</td>
</tr>
<tr>
<td>NOCLPRINT</td>
<td>Suppresses “Class Level Information” completely or in parts</td>
</tr>
<tr>
<td>NOITPRINT</td>
<td>Suppresses “Iteration History” table</td>
</tr>
<tr>
<td>RANKS</td>
<td>Displays a table of ranks of matrices X, (XZ), and MMEQ</td>
</tr>
<tr>
<td>SIMPLE</td>
<td>Displays “Descriptive Statistics” table</td>
</tr>
</tbody>
</table>

#### Singularity Tolerances

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SINGCHOL=</td>
<td>Tunes singularity for Cholesky decompositions</td>
</tr>
<tr>
<td>SINGRES=</td>
<td>Tunes singularity for the residual variance</td>
</tr>
<tr>
<td>SINGULAR=</td>
<td>Tunes general singularity criterion</td>
</tr>
</tbody>
</table>

You can specify the following options.

**BLUP**( <suboptions> ) = SAS-data-set

creates a data set that contains the BLUE and BLUP solutions. The covariance parameters are assumed to be known and given by PARMS statement. All hypothesis testing is ignored. The statements TEST, ESTIMATE, CONTRAST, LSMEANS, and OUTPUT are all ignored. This option is designed for users who need BLUP solutions for random effects with many levels, up to tens of millions.

You can specify the following suboptions:

**ITPRINT=** *number* specifies that the iteration history be displayed after every *number* of iterations. This suboption applies only for iterative solving methods (IOC or IOD). The default value is 10, which means the procedure displays the iteration history for every 10 iterations.

**MAXITER=** *number* specifies the maximum number of iterations allowed. This applies only for iterative solving methods (IOC or IOD). The default value is the number of parameters in the BLUE/BLUP plus two.

**METHOD=** DIRECT | IOC | IOD specifies the method used to solve for BLUP solutions. **METHOD=** DIRECT requires storing mixed model equations (MMEQ) in memory and computing the Cholesky decomposition of MMEQ. This method is the most accurate, but it is the most inefficient in terms of speed and memory. **METHOD=** IOD does not build mixed model equations; instead it iterates on data to solve for the solutions. This method is most efficient in terms of memory. **METHOD=** IOC requires storing mixed model equations in memory and iterates on MMEQ to solve for the solutions. This method is the most efficient in terms of speed. The default method is IOC.
**TOL=number** specifies the tolerance value. This *suboption* applies only for iterative solving methods (IOC or IOD). The default value is the square root of machine precision.

**DATA=SAS-data-set**

names the SAS data set to be used by PROC HPMIXED. The default is the most recently created data set.

**INFCRIT=NONE | PQ | Q**

**IC=NONE | PQ | Q**

determines the computation of information criteria in the “Fit Statistics” table. The criteria are all in smaller-is-better form, and are described in *Table 49.3*.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Formula</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIC</td>
<td>(-2\ell + 2d)</td>
<td>Akaike (1974)</td>
</tr>
<tr>
<td>AICC</td>
<td>(-2\ell + 2dn^<em>(n^</em> - d - 1)) for (n^* \geq d + 2)</td>
<td>Hurvich and Tsai (1989) and Burnham and Anderson (1998)</td>
</tr>
<tr>
<td></td>
<td>(-2\ell + 2d(d + 2)) for (n^* &lt; d + 2)</td>
<td></td>
</tr>
<tr>
<td>HQIC</td>
<td>(-2\ell + 2d \log(\log(n))) for (n &gt; 1)</td>
<td>Hannan and Quinn (1979)</td>
</tr>
<tr>
<td>BIC</td>
<td>(-2\ell + d \log(n)) for (n &gt; 0)</td>
<td>Schwarz (1978)</td>
</tr>
<tr>
<td>CAIC</td>
<td>(-2\ell + d(\log(n) + 1)) for (n &gt; 0)</td>
<td>Bozdogan (1987)</td>
</tr>
</tbody>
</table>

Here \(\ell\) denotes the maximum value of the restricted log likelihood, \(d\) is the dimension of the model, and \(n, n^*\) reflect the size of the data. When \(n \leq 1\), the value of the HQIC criterion is \(-2\ell\). When \(n=0\), the values of the BIC and CAIC criteria are undefined.

The quantities \(d, n, n^*\) depend on the model and IC= option.

- models without random effects:
  - The IC=Q and IC=PQ options have no effect on the computation.
  - \(d\) equals the number of parameters in the optimization whose solutions do not fall on the boundary or are otherwise constrained.
  - \(n\) equals the number of used observations minus rank(\(X\)).
  - \(n^*\) equals \(n\), unless \(n < d + 2\), in which case \(n^* = d + 2\).

- models with random effects:
  - \(d\) equals the number of parameters in the optimization whose solutions do not fall on the boundary or are otherwise constrained. If IC=PQ, this value is incremented by rank(\(X\)).
  - \(n\) equals the effective number of subjects as displayed in the “Dimensions” table, unless this value equals 1, in which case \(n\) equals the number of levels of the first random effect specified. The IC=Q and IC=PQ options have no effect.
  - \(n^*\) equals \(n\), unless \(n < d + 2\), in which case \(n^* = d + 2\). The IC=Q and IC=PQ options have no effect.

The IC=NONE option suppresses the “Fit Statistics” table. IC=Q is the default.
ITDETAILS
displays the parameter values at each iteration and enables the writing of notes to the SAS log pertaining to “infinite likelihood” and “singularities” during optimization iterations.

LOGNOTE
writes to the log periodic notes that describe the current status of computations. This option is designed for use with analyses that require extensive CPU resources.

MAXCLPRINT=number
specifies the maximum levels of CLASS variables to print in the ODS table ClassLevels. The default value is 20. MAXCLPRINT=0 enables you to print all levels of each CLASS variable. However, the option NOCLPRINT takes precedence over MAXCLPRINT.

METHOD=
specifies the estimation method for the covariance parameters. The REML specification performs residual (restricted) maximum likelihood, and it is currently the only available method. This option is therefore currently redundant for PROC HPMIXED, but it is included for consistency with other mixed model procedures in SAS/STAT software.

MMEQ
displays coefficients of the mixed model equations. These are

\[
\begin{bmatrix}
X^'R^{-1}X & X^'R^{-1}Z \\
Z^'R^{-1}X & Z^'R^{-1}Z + \hat{G}^{-1}
\end{bmatrix}
\begin{bmatrix}
X^'R^{-1}y \\
Z^'R^{-1}y
\end{bmatrix}
\]

assuming \(\hat{G}\) is nonsingular. If \(\hat{G}\) is singular, PROC HPMIXED produces the following coefficients

\[
\begin{bmatrix}
X^'R^{-1}X & X^'R^{-1}Z\hat{G} \\
\hat{G}Z^'R^{-1}X & \hat{G}Z^'R^{-1}Z\hat{G} + \hat{G}
\end{bmatrix}
\begin{bmatrix}
X^'R^{-1}y \\
\hat{G}Z^'R^{-1}y
\end{bmatrix}
\]

See the section “Model and Assumptions” on page 3857 for further information about these equations.

NAMELEN=number
specifies the length to which long effect names are shortened. The default and minimum value is 20.

NLPRINT
requests that optimization-related output options specified in the NLOPTIONS statement override corresponding options in the PROC HPMIXED statement. When you specify NLPRINT, the ITDETAILS and NOITPRINT options in the PROC HPMIXED statement are ignored and the following six options in the NLOPTIONS statement are enabled: NOPRINT, PHISTORY, PSUMMARY, PALL, PLONG, and PHISTPARMS.

The syntax and options of the NLOPTIONS statement are described in the section “NLOPTIONS Statement” on page 488 in Chapter 19, “Shared Concepts and Topics.”

NOCLPRINT<=number>
suppresses the display of the “Class Level Information” table if you do not specify number. If you do specify number, only levels with totals that are less than number are listed in the table.
NOFIT
suppresses fitting of the model. When the NOFIT option is in effect, PROC HPMIXED produces the “Model Information,” “Class Level Information,” “Number of Observations,” “Dimensions,” and “Descriptive Statistics” tables. These can be helpful in gauging the computational effort required to fit the model.

NOINFO
suppresses the display of the “Model Information,” “Number of Observations,” and “Dimensions” tables.

NOITPRINT
suppresses the display of the “Iteration History” table.

NOPRINT
suppresses the normal display of results. The NOPRINT option is useful when you want only to create one or more output data sets with the procedure by using the OUTPUT statement. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, “Using the Output Delivery System,” for more information.

NOPROFILE
includes the residual variance as one of the covariance parameters in the optimization iterations. This option applies only to models that have a residual variance parameter. By default, this parameter is profiled out of the optimization iterations, except when you have specified the HOLD= option in the PARMS statement.

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the sort order for the levels of the classification variables (which are specified in the CLASS statement).

This option applies to the levels for all classification variables, except when you use the (default) ORDER=FORMATTED option with numeric classification variables that have no explicit format. In that case, the levels of such variables are ordered by their internal value.

The ORDER= option can take the following values:

<table>
<thead>
<tr>
<th>Value of 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 with no explicit format, which are sorted by their unformatted (internal) value</td>
</tr>
<tr>
<td>FREQ</td>
<td>Descending frequency count; levels with the most observations come first in the order</td>
</tr>
<tr>
<td>INTERNAL</td>
<td>Unformatted value</td>
</tr>
</tbody>
</table>

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

For more information about sort order, see the chapter on the SORT procedure in the Base SAS Procedures Guide and the discussion of BY-group processing in SAS Language Reference: Concepts.
RANKS
displays the ranks of design matrices $X$ and $(XZ)$ and the coefficient matrix of the mixed model equations (MMEQ).

SIMPLE
displays the mean, standard deviation, coefficient of variation, minimum, and maximum for each variable used in PROC HPMIXED that is not a classification variable.

SINGCHOL=number
tunes the singularity criterion in Cholesky decompositions. The default is $1E6$ times the machine epsilon; this product is approximately $1E–10$ on most computers.

SINGRES=number
sets the tolerance for which the residual variance is considered to be zero. The default is $1E4$ times the machine epsilon; this product is approximately $1E–12$ on most computers.

SINGULAR=number
tunes the general singularity criterion applied by the HPMIXED procedure in divisions and inversions. The default is $1E4$ times the machine epsilon; this product is approximately $1E–12$ on most computers.

UPDATE
is an alias for the LOGNOTE option.

---

**BY Statement**

**BY** variables ;

You can specify a BY statement with PROC HPMIXED to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the HPMIXED procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

Since sorting the data changes the order in which PROC HPMIXED reads observations, the sort order for the levels of the CLASS variable might be affected if you have specified ORDER=DATA in the PROC HPMIXED statement. This, in turn, affects specifications in the CONTRAST and ESTIMATE statements.

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*. 
The CLASS statement names the classification variables to be used in the model. Typical classification variables are Treatment, Sex, Race, Group, and Replication. If you use the CLASS statement, it must appear before the MODEL statement.

Classification variables can be either character or numeric. By default, class levels are determined from the entire set of formatted values of the CLASS variables.

**NOTE:** Prior to SAS 9, class levels were determined by using no more than the first 16 characters of the formatted values. To revert to this previous behavior, you can use the TRUNCATE option in the CLASS statement.

In any case, you can use formats to group values into levels. See the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and the discussions of the FORMAT statement and SAS formats in *SAS Formats and Informats: Reference*. You can adjust the order of CLASS variable levels with the ORDER= option in the PROC HPMIXED statement.

You can specify the following REF= option to indicate how the levels of an individual classification variable are to be ordered by enclosing it in parentheses after the variable name:

**REF=’level’ | FIRST | LAST**

specifies a level of the classification variable to be put at the end of the list of levels. (In procedures that solve mixed model equations by sequentially sweeping rows and columns, this level thus corresponds to the reference level in the usual interpretation of the estimates of a singular parameterization. However, since PROC HPMIXED does not necessarily solve mixed model equations in the original order, this interpretation of the specified REF= level does not apply for this procedure.) You can specify the level of the variable to use as the reference level; specify a value that corresponds to the formatted value of the variable if a format is assigned. Alternatively, you can specify REF=FIRST to designate that the first ordered level serve as the reference, or REF=LAST to designate that the last ordered level serve as the reference. To specify that REF=FIRST or REF=LAST be used for all classification variables, use the REF= global-option after the slash (/) in the CLASS statement.

You can specify the following global-options in the CLASS statement after a slash (/):

**REF=FIRST | LAST**

specifies a level of all classification variables to be put at the end of the list of levels. (In procedures that solve mixed model equations by sequentially sweeping rows and columns, this level thus corresponds to the reference level in the usual interpretation of the estimates of a singular parameterization. However, since PROC HPMIXED does not necessarily solve mixed model equations in the original order, this interpretation of the specified REF= level does not apply for this procedure.) Specify REF=FIRST to designate that the first ordered level serve as the reference. Specify REF=LAST to designate that the last ordered level serve as the reference. This option applies to all the variables specified in the CLASS statement. To specify different reference levels for different classification variables, use REF= options for individual variables.
TRUNCATE specifies that class levels be determined by using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option to revert to the levels as determined in releases prior to SAS 9.

CONTRAST Statement

CONTRAST 'label' contrast-specification <, contrast-specification> <, ... > < / options> ;

The CONTRAST statement provides a mechanism for obtaining custom hypothesis tests. It is patterned after the CONTRAST statement in PROC MIXED and enables you to select an appropriate inference space (McLean, Sanders, and Stroup 1991).

You can test the hypothesis \( \mathbf{L}' \mathbf{\phi} = 0 \), where \( \mathbf{L}' = [\mathbf{K}' \mathbf{M}'] \) and \( \mathbf{\phi}' = [\mathbf{\beta}' \mathbf{\gamma}'] \), in several inference spaces. The inference space corresponds to the choice of \( \mathbf{M} \). When \( \mathbf{M} = 0 \), your inferences apply to the entire population from which the random effects are sampled; this is known as the broad inference space. When all elements of \( \mathbf{M} \) are nonzero, your inferences apply only to the observed levels of the random effects. This is known as the narrow inference space, and you can also choose it by specifying all of the random effects as fixed. The GLM procedure uses the narrow inference space. Finally, by zeroing portions of \( \mathbf{M} \) corresponding to selected main effects and interactions, you can choose intermediate inference spaces. The broad inference space is usually the most appropriate, and it is used when you do not specify any random effects in the CONTRAST statement.

In the CONTRAST statement,

- **label** identifies the contrast in the table. A label is required for every contrast specified. Labels can be up to 20 characters and must be enclosed in single quotes.
- **contrast-specification** identifies the fixed effects and random effects and their coefficients from which the \( \mathbf{L} \) matrix is formed. The syntax representation of a **contrast-specification** is < fixed-effect values . . . > | < fixed-effect values . . . >
- **fixed-effect** identifies an effect that appears in the MODEL statement. The keyword INTERCEPT can be used as an effect when an intercept is fitted in the model. You do not need to include all effects that are in the MODEL statement.
- **random-effect** identifies an effect that appears in the RANDOM statement. The first random effect must follow a vertical bar (|); however, random effects do not have to be specified.
- **values** are constants that are elements of the \( \mathbf{L} \) matrix associated with the fixed and random effects.

The rows of \( \mathbf{L}' \) are specified in order and are separated by commas. The rows of the \( \mathbf{K}' \) component of \( \mathbf{L}' \) are specified on the left side of the vertical bars (|). These rows test the fixed effects and are, therefore, checked for estimability. The rows of the \( \mathbf{M}' \) component of \( \mathbf{L}' \) are specified on the right side of the vertical bars. They test the random effects, and no estimability checking is necessary.

If PROC HPMIXED finds the fixed-effects portion of the specified contrast to be nonestimable (see the SINGULAR= option on page 3835), then it displays missing values for the test statistics and a note in the log.
If the elements of \( L \) are not specified for an effect that contains a specified effect, then the elements of the specified effect are automatically “filled in” over the levels of the higher-order effect. This feature is designed to preserve estimability for cases where there are complex higher-order effects. The coefficients for the higher-order effect are determined by equitably distributing the coefficients of the lower-level effect as in the construction of least squares means. In addition, if the intercept is specified, it is distributed over all classification effects that are not contained by any other specified effect. If an effect is not specified and does not contain any specified effects, then all of its coefficients in \( L \) are set to 0. You can override this behavior by specifying coefficients for the higher-order effect.

If too many values are specified for an effect, the extra ones are ignored; if too few are specified, the remaining ones are set to 0. If no random effects are specified, the vertical bar can be omitted; otherwise, it must be present. If a SUBJECT effect is used in the RANDOM statement, then the coefficients specified for the effects in the RANDOM statement are equitably distributed across the levels of the SUBJECT effect. You can use the E option to see exactly what \( L \) matrix is used.

The SUBJECT and GROUP options in the CONTRAST statement are useful for the case where a SUBJECT= or GROUP= variable appears in the RANDOM statement, and you want to contrast different subjects or groups. By default, CONTRAST statement coefficients about random effects are distributed equally across subjects and groups.

PROC HPMIXED handles missing level combinations of CLASS variables similarly to the way PROC GLM does. Both procedures delete fixed-effects parameters corresponding to missing levels in order to preserve estimability. However, PROC HPMIXED does not delete missing level combinations for random-effects parameters because linear combinations of the random-effects parameters are always estimable. These conventions can affect the way you specify your CONTRAST coefficients.

The CONTRAST statement computes the statistic

\[
F = \frac{\left( \begin{array}{c} \hat{\beta} \\ \hat{\gamma} \end{array} \right)' \left( L(L'\hat{C}L)^{-1}L' \right) \left( \begin{array}{c} \hat{\beta} \\ \hat{\gamma} \end{array} \right)}{r}
\]

where \( r = \text{rank}(L'\hat{C}L) \) and approximates its distribution with an \( F \) distribution. In this expression, \( \hat{C} \) is an estimate of the generalized inverse of the coefficient matrix in the mixed model equations.

The numerator degree of freedom in the \( F \) approximation is \( r = \text{rank}(L'\hat{C}L) \), and the denominator degree of freedom is taken from the “Type III Tests of Fixed Effects” table and corresponds to the final effect you list in the CONTRAST statement. You can change the denominator degrees of freedom by using the DF= option.

You can specify the following options in the CONTRAST statement after a slash (/).

**CHISQ**

requests that \( \chi^2 \) tests be performed in addition to any \( F \) tests. A \( \chi^2 \) statistic equals its corresponding \( F \) statistic times the associate numerator degree of freedom, and this same degree of freedom is used to compute the \( p \)-value for the \( \chi^2 \) test. This \( p \)-value will always be less than that for the \( F \) test, as it effectively corresponds to an \( F \) test with infinite denominator degrees of freedom.

**DF=number**

specifies the denominator degrees of freedom for the \( F \) test. The default is the denominator degrees of freedom taken from the “Type III Tests of Fixed Effects” table and corresponds to the final effect you list in the CONTRAST statement.
requests that the L matrix coefficients for the contrast be displayed. The name of this “L Matrix Coefficients” table is “Coef.”

GROUP coeffs
sets up random-effect contrasts between different groups when a GROUP= variable appears in the RANDOM statement. By default, CONTRAST statement coefficients about random effects are distributed equally across groups. If you enter a multi-row contrast, you can also enter multiple rows for the GROUP coefficients. If the number of GROUP coefficients is less than the number of contrasts in the CONTRAST statement, the HPMIXED procedure cycles through the GROUP coefficients. For example, the following two statements are equivalent:

contrast 'Trt @ x=0.4 and 0.5' trt 1 -1 0 | x 0.4,
   trt 1 0 -1 | x 0.4,
   trt 1 -1 0 | x 0.5,
   trt 1 0 -1 | x 0.5 /
   group 1 -1, 1 0 -1, 1 -1, 1 0 -1;

contrast 'Trt @ x=0.4 and 0.5' trt 1 -1 0 | x 0.4,
   trt 1 0 -1 | x 0.4,
   trt 1 -1 0 | x 0.5,
   trt 1 0 -1 | x 0.5 /
   group 1 -1, 1 0 -1;

SINGULAR=number
tunes the estimability checking. If v is a vector, define ABS(v) to be the largest absolute value of the element of v with the largest absolute value. If ABS(K'HK) is greater than c*number for any row of K' in the contrast, then K is declared nonestimable. Here T is the Hermite form matrix (X'X)−1X'X, and c is ABS(K') except when it equals 0, and then c is 1. The value for number must be between 0 and 1; the default is 1E−4.

SUBJECT coeffs
sets up random-effect contrasts between different subjects when a SUBJECT= variable appears in the RANDOM statement. By default, CONTRAST statement coefficients about random effects are distributed equally across subjects. Listing subject coefficients for multiple row CONTRASTS follows the same rules as for GROUP coefficients.
COLLECTION is a collection effect that defines one or more variables as a single effect with multiple degrees of freedom. The variables in a collection are considered as a unit for estimation and inference.

LAG is a classification effect in which the level that is used for a given period corresponds to the level in the preceding period.

MULTIMEMBER | MM is a multimember classification effect whose levels are determined by one or more variables that appear in a CLASS statement.

POLYNOMIAL | POLY is a multivariate polynomial effect in the specified numeric variables.

SPLINE is a regression spline effect whose columns are univariate spline expansions of one or more variables. A spline expansion replaces the original variable with an expanded or larger set of new variables.

Table 49.4 summarizes the options available in the EFFECT statement.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Collection Effects Options</strong></td>
<td></td>
</tr>
<tr>
<td>DETAILS</td>
<td>Displays the constituents of the collection effect</td>
</tr>
<tr>
<td><strong>Lag Effects Options</strong></td>
<td></td>
</tr>
<tr>
<td>DESIGNROLE=</td>
<td>Names a variable that controls to which lag design an observation is assigned</td>
</tr>
<tr>
<td>DETAILS</td>
<td>Displays the lag design of the lag effect</td>
</tr>
<tr>
<td>NLAG=</td>
<td>Specifies the number of periods in the lag</td>
</tr>
<tr>
<td>PERIOD=</td>
<td>Names the variable that defines the period</td>
</tr>
<tr>
<td>WITHIN=</td>
<td>Names the variable or variables that define the group within which each period is defined</td>
</tr>
<tr>
<td><strong>Multimember Effects Options</strong></td>
<td></td>
</tr>
<tr>
<td>NOEFFECT</td>
<td>Specifies that observations with all missing levels for the multimember variables should have zero values in the corresponding design matrix columns</td>
</tr>
<tr>
<td>WEIGHT=</td>
<td>Specifies the weight variable for the contributions of each of the classification effects</td>
</tr>
<tr>
<td><strong>Polynomial Effects Options</strong></td>
<td></td>
</tr>
<tr>
<td>DEGREE=</td>
<td>Specifies the degree of the polynomial</td>
</tr>
<tr>
<td>MDEGREE=</td>
<td>Specifies the maximum degree of any variable in a term of the polynomial</td>
</tr>
<tr>
<td>STANDARDIZE=</td>
<td>Specifies centering and scaling suboptions for the variables that define the polynomial</td>
</tr>
</tbody>
</table>
Table 49.4  continued

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline Effects Options</td>
<td></td>
</tr>
<tr>
<td>BASIS=</td>
<td>Specifies the type of basis (B-spline basis or truncated power function basis) for the spline effect</td>
</tr>
<tr>
<td>DEGREE=</td>
<td>Specifies the degree of the spline effect</td>
</tr>
<tr>
<td>KNOTMETHOD=</td>
<td>Specifies how to construct the knots for the spline effect</td>
</tr>
</tbody>
</table>

For more information about the syntax of these *effect-types* and how columns of constructed effects are computed, see the section “EFFECT Statement” on page 397 in Chapter 19, “Shared Concepts and Topics.”

The HPMIXED procedure does not support the SPLIT or SEPARATED option in spline effects and poly effects.

**ESTIMATE Statement**

```plaintext
ESTIMATE 'label' contrast-specification < (divisor=n) >
< , 'label' contrast-specification < (divisor=n) > < , . . . > </ options > ;
```

The ESTIMATE statement provides a mechanism for obtaining custom hypothesis tests. As in the CONTRAST statement, the basic element of the ESTIMATE statement is the *contrast-specification*, which consists of MODEL and RANDOM effects and their coefficients. Specifically, a *contrast-specification* takes the form

```
< fixed-effect values . . . > < | random-effect values . . . >
```

Based on the *contrast-specifications* in your ESTIMATE statement, PROC HPMIXED constructs the matrix $L' = [K' M']$, as in the CONTRAST statement, where $K$ is associated with the fixed effects and $M$ is associated with the G-side random effects.

PROC HPMIXED then produces for each row $l$ of $L'$ an approximate $t$ test of the hypothesis $H: l\phi = 0$, where $\phi = [\beta' \gamma']'$. Results from all ESTIMATE statement are combined in the “Estimates” ODS table.

Note that multi-row estimates are permitted. Unlike the CONTRAST statement, you need to specify a *'label'* for every row of the multi-row estimate, since PROC HPMIXED produces one test per row.

PROC HPMIXED selects the degrees of freedom to match those displayed in the “Type III Tests of Fixed Effects” table for the final effect you list in the ESTIMATE statement. You can modify the degrees of freedom by using the DF= option. If you select DDFM=NONE and do not modify the degrees of freedom by using the DF= option, PROC HPMIXED uses infinite degrees of freedom, essentially computing approximate $z$ tests.

If PROC HPMIXED finds the fixed-effects portion of the specified estimate to be nonestimable, then it displays “Non-est” for the estimate entry.

The construction of the $L$ matrix for an ESTIMATE statement follows the same rules as listed under the CONTRAST statement.
Table 49.5 summarizes the options available in the ESTIMATE statement.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA=</td>
<td>Specifies the confidence level</td>
</tr>
<tr>
<td>CL</td>
<td>Constructs t-type confidence limits</td>
</tr>
<tr>
<td>DF=</td>
<td>Specifies the degrees of freedom</td>
</tr>
<tr>
<td>DIVISOR=</td>
<td>Specifies values to divide the coefficients</td>
</tr>
<tr>
<td>E</td>
<td>Displays the matrix coefficients</td>
</tr>
<tr>
<td>GROUP</td>
<td>Sets up random-effect contrasts between groups</td>
</tr>
<tr>
<td>SINGULAR=</td>
<td>Tunes the estimability checking</td>
</tr>
<tr>
<td>SUBJECT</td>
<td>Sets up random-effect estimates between subjects</td>
</tr>
</tbody>
</table>

You can specify the following options in the ESTIMATE statement after a slash (/).

**ALPHA=** *number*

requests that a t-type confidence interval be constructed with confidence level $1 - \text{number}$. The value of *number* must be between 0 and 1 exclusively; the default is 0.05. If DDFM=NONE and you do not specify degrees of freedom with the DF= option, PROC HPMIXED uses infinite degrees of freedom, essentially computing a z interval.

**CL**

requests that t-type confidence limits be constructed. If DDFM=NONE and you do not specify degrees of freedom with the DF= option, PROC HPMIXED uses infinite degrees of freedom, essentially computing a z interval. The confidence level is 0.95 by default.

**DF=** *number*

specifies the degrees of freedom for the t-test. The default is the denominator degrees of freedom taken from the “Type III Tests of Fixed Effects” table and corresponds to the final effect you list in the ESTIMATE statement.

**DIVISOR=** *value-list*

specifies a list of values by which to divide the coefficients so that fractional coefficients can be entered as integer numerators. If you do not specify *value-list*, a default value of 1.0 is assumed. Missing values in the *value-list* are converted to 1.0.

If the number of elements in *value-list* exceeds the number of rows of the estimate, the extra values are ignored. If the number of elements in *value-list* is less than the number of rows of the estimate, the last value in *value-list* is copied forward.

If you specify a row-specific divisor as part of the specification of the estimate row, this value multiplies the corresponding divisor implied by the *value-list*. For example, the following statement divides the coefficients in the first row by 8, and the coefficients in the third and fourth row by 3:

```
estimate 'One vs. two' A 2 -2 (divisor=2),
    'One vs. three' A 1 0 -1 ,
    'One vs. four' A 3 0 0 -3 ,
    'One vs. five' A 1 0 0 0 -1 / divisor=4,.,3;
```
requests that the matrix coefficients be displayed. For ODS purposes, the name of this “L Matrix Coefficients” table is “Coef.”

**GROUP coeffs**

sets up random-effect contrasts between different groups when a GROUP= variable appears in the RANDOM statement. By default, ESTIMATE statement coefficients about random effects are distributed equally across groups. If you enter a multi-row estimate, you can also enter multiple rows for the GROUP coefficients. If the number of GROUP coefficients is less than the number of contrasts in the ESTIMATE statement, the HPMIXED procedure cycles through the GROUP coefficients. For example, the following two statements are equivalent:

```plaintext
estimate 'Trt 1 vs 2 @ x=0.4' trt 1 -1 0 | x 0.4,
  'Trt 1 vs 3 @ x=0.4' trt 1 0 -1 | x 0.4,
  'Trt 1 vs 2 @ x=0.5' trt 1 -1 0 | x 0.5,
  'Trt 1 vs 3 @ x=0.5' trt 1 0 -1 | x 0.5 /
    group 1 -1, 1 0 -1, 1 -1, 1 0 -1;
```

```plaintext
estimate 'Trt 1 vs 2 @ x=0.4' trt 1 -1 0 | x 0.4,
  'Trt 1 vs 3 @ x=0.4' trt 1 0 -1 | x 0.4,
  'Trt 1 vs 2 @ x=0.5' trt 1 -1 0 | x 0.5,
  'Trt 1 vs 3 @ x=0.5' trt 1 0 -1 | x 0.5 /
    group 1 -1, 1 0 -1;
```

**SINGULAR=number**

tunes the estimability checking as documented for the SINGULAR= in the CONTRAST statement.

**SUBJECT coeffs**

sets up random-effect estimates between different subjects when a SUBJECT= variable appears in the RANDOM statement. By default, ESTIMATE statement coefficients about random effects are distributed equally across subjects. Listing subject coefficients for an ESTIMATE statement with multiple rows follows the same rules as for GROUP coefficients.

### ID Statement

**ID variables ;**

The ID statement specifies which variables from the input data set are to be included in the OUT= data sets from the OUTPUT statement. If you do not specify an ID statement, then all variables are included in these data sets. Otherwise, only the variables you list in the ID statement are included. Specifying an ID statement with no variables prevents any variables from being included in these data sets.

### LSMEANS Statement

**LSMEANS fixed-effects </options> ;**
The LSMEANS statement computes least squares means (LS-means) of fixed effects. As in the GLM procedure, LS-means are *predicted population margins*—that is, they estimate the marginal means over a balanced population. In a sense, LS-means are to unbalanced designs as classification and subclassification arithmetic means are to balanced designs. The $L$ matrix constructed to compute them is the same as the $L$ matrix formed in PROC GLM; however, the standard errors are adjusted for the covariance parameters in the model.

Each LS-mean is computed as $L\hat{\beta}$, where $L$ is the coefficient matrix associated with the least squares mean and $\hat{\beta}$ is the estimate of the fixed-effects parameter vector. The approximate standard errors for the LS-mean is computed as the square root of $L' (X'V^{-1}X)^{-1} L$.

LS-means can be computed for any effect in the MODEL statement that involves CLASS variables. You can specify multiple effects in one LSMEANS statement or in multiple LSMEANS statements, and all LSMEANS statements must appear after the MODEL statement. As in the ESTIMATE statement, the $L$ matrix is tested for estimability, and if this test fails, PROC HPMIXED displays “Non-est” for the LS-means entries.

Assuming the LS-mean is estimable, PROC HPMIXED constructs an approximate $t$ test to test the null hypothesis that the associated population quantity equals zero. By default, the denominator degrees of freedom for this test are the same as those displayed for the effect in the “Type III Tests of Fixed Effects” table (see the section “TEST Statement” on page 3856).

Table 49.6 summarizes the options available in the LSMEANS statement.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>Specifies the confidence level</td>
</tr>
<tr>
<td>CL</td>
<td>Constructs $t$-type confidence limits</td>
</tr>
<tr>
<td>CORR</td>
<td>Displays the estimated correlation matrix</td>
</tr>
<tr>
<td>COV</td>
<td>Displays the estimated covariance matrix</td>
</tr>
<tr>
<td>DF</td>
<td>Specifies the degrees of freedom</td>
</tr>
<tr>
<td>DIFF or PDIFF</td>
<td>Displays the differences of the LS-means</td>
</tr>
<tr>
<td>E</td>
<td>Displays the matrix coefficients for LSMEANS effects</td>
</tr>
<tr>
<td>SINGULAR</td>
<td>Tunes the estimability checking</td>
</tr>
<tr>
<td>SLICE</td>
<td>Partitions interaction LSMEANS effects</td>
</tr>
</tbody>
</table>

You can specify the following options in the LSMEANS statement after a slash (/).

**ALPHA=** *number*  
requests that a $t$-type confidence interval be constructed for each of the LS-means with confidence level $1 - number$. The value of *number* must be between 0 and 1; the default is 0.05.

**CL**  
requests that $t$-type confidence limits be constructed for each of the LS-means. If DDFM=NONE, then PROC HPMIXED uses infinite degrees of freedom for this test, essentially computing a $z$ interval. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.
**LSMEANS Statement**

**CORR**

displays the estimated correlation matrix of the least squares means as part of the “Least Squares Means” table.

**COV**

displays the estimated covariance matrix of the least squares means as part of the “Least Squares Means” table.

**DF=** number

specifies the degrees of freedom for the $t$ test and confidence limits. The default is the denominator degrees of freedom taken from the “Type III Tests of Fixed Effects” table corresponding to the LS-means effect. For these DDFM= methods, degrees of freedom are determined separately for each test; see the DDFM= option on page 3843 for more information.

**DIFF<=difftype>**

**PDIFF<=difftype>**

requests that differences of the LS-means be displayed. You can specify the following values for the optional difftype.

- **ALL** requests all pairwise differences; it is the default.
- **ANOM** requests differences between each LS-mean and the average LS-mean, as in the analysis of means (Ott 1967). The average is computed as a weighted mean of the LS-means, with the weights being inversely proportional to the diagonal entries of the $L(X'X)^{-1}L'$ matrix. When a WEIGHT statement is specified, then the preceding matrix is replaced with $L(X'WX)^{-1}L'$ where $W$ is the diagonal matrix that contains the weights. If LS-means are nonestimable, this design-based weighted mean is replaced with an equally weighted mean. Note that the ANOM procedure in SAS/QC software implements both tables and graphics for the analysis of means with a variety of response types. For one-way designs and normally distributed data, the DIFF=ANOM computations are equivalent to the results of PROC ANOM.
- **CONTROL** requests differences with a control; by default, the control is the first level of each of the specified LSMEANS effects. To specify which levels of the effects are the controls, list the quoted formatted values in parentheses after the CONTROL keyword. For example, if the effects A, B, and C are classification variables, each having two levels, 1 and 2, the following LSMEANS statement specifies the (1,2) level of A*B and the (2,1) level of B*C as controls:

  ```plaintext
  lsmeans A*B B*C / diff=control('1' '2' '2' '1');
  ```

  For multiple effects, the results depend upon the order of the list, and so you should check the output to make sure that the controls are correct. CONTROL produces two-tailed tests and confidence limits.

- **CONTROLL** requests one-tailed results and tests whether the noncontrol levels are significantly smaller than the control. The upper confidence limits for the control minus the noncontrol levels are considered to be infinity and are displayed as missing.

- **CONTROLU** requests one-tailed results and tests whether the noncontrol levels are significantly larger than the control. The upper confidence limits for the noncontrol levels minus the control are considered to be infinity and are displayed as missing.
Chapter 49: The HPMIXED Procedure

The differences of the LS-means are displayed in a table titled “Differences of Least Squares Means.” The table name is “Diffs.”

E
requests that the matrix coefficients for all LSMEANS effects be displayed. The name of this “Matrix Coefficients” table is “Coef.”

PDIFF
is the same as the DIFF option. See the description of the DIFF option on page 3841.

SINGULAR=number

tunes the estimability checking as documented for the SINGULAR= in the CONTRAST statement.

SLICE=fixed-effect | (fixed-effects)
specifies effects by which to partition interaction LSMEANS effects. This can produce what are known as tests of simple effects (Winer 1971). For example, suppose that A*B is significant, and you want to test the effect of A for each level of B. The appropriate LSMEANS statement is

   lsmeans A*B / slice=B;

This statement tests for the simple main effects of A for B, which are calculated by extracting the appropriate rows from the coefficient matrix for the A*B LS-means and by using them to form an F test.

The SLICE= option produces F tests that test the simultaneous equality of cell means at a fixed level of the slice effect (Schabenberger, Gregoire, and Kong 2000).

The SLICE= option produces a table titled “Tests of Effect Slices.” The table name is “Slices.”

MODEL Statement

   MODEL dependent = < fixed-effects > / < options > ;

The MODEL statement names a single dependent variable and the fixed effects, which determine the X matrix of the mixed model. The specification of effects is the same as in the GLM procedure; however, unlike PROC GLM, you do not specify random effects in the MODEL statement. The MODEL statement is required.

An intercept is included in the fixed-effects model by default. If no fixed effects are specified, only this intercept term is fit. The intercept can be removed by using the NOINT option.

You can specify the following options in the MODEL statement after a slash (/).

   ALPHA=number
   requests that a t-type confidence interval be constructed for each of the fixed-effects parameters with confidence level 1 − number. The value of number must be between 0 and 1; the default is 0.05.

   CL
   requests that t-type confidence limits be constructed for each of the fixed-effects parameter estimates. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.
**DDF=value-list**

enables you to specify your own denominator degrees of freedom for the fixed effects. The *value-list* specification is a list of numbers or missing values (.) separated by commas. The degrees of freedom should be listed in the order in which the effects appear in the “Type III Tests of Fixed Effects” table. If you want to retain the default degrees of freedom for a particular effect, use a missing value for its location in the list. For example, the following statement assigns 3 denominator degrees of freedom to A and 4.7 to A*B, while those for B remain the same:

```plaintext
model Y = A B A*B / ddf=3,.,4.7;
```

**DDFM=RESIDUAL | NONE**

specifies the method for computing the denominator degrees of freedom for the tests of fixed effects resulting from the MODEL, CONTRAST, ESTIMATE, LSMEANS, and TEST statements.

The DDFM=RESIDUAL option performs all tests by using the residual degrees of freedom, \( n - \text{rank}(X) \), where \( n \) is the number of observations used. It is the default degrees of freedom method.

DDFM=NONE specifies that no denominator degrees of freedom be applied. PROC HPMIXED then essentially assumes that infinite degrees of freedom are available in the calculation of \( p \)-values. The \( p \)-values for \( t \) tests are then identical to \( p \)-values derived from the standard normal distribution. In the case of \( F \) tests, the \( p \)-values equal those of chi-square tests determined as follows: if \( F_{obs} \) is the observed value of the \( F \) test with \( l \) numerator degrees of freedom, then

\[
p = \Pr\{F_{l,\infty} > F_{obs}\} = \Pr\{\chi_{l}^{2} > l F_{obs}\}
\]

**NOINT**

requests that no intercept be included in the model. An intercept is included by default.

**SOLUTION | S**

requests that a solution for the fixed-effects parameters be produced. Using notation from the section “Model Assumptions” on page 3857, the fixed-effects parameter estimates are \( \hat{\beta} \) and their approximate standard errors are the square roots of the diagonal elements of \( X^{\prime} \hat{V}^{-1} X \). Along with the estimates and their approximate standard errors, a \( t \) statistic is computed as the estimate divided by its standard error. The degree of freedom for this \( t \) statistic matches the one appearing in the “Type III Tests of Fixed Effects” table under the effect containing the parameter. The “Pr > |t|” column contains the two-tailed \( p \)-value corresponding to the \( t \) statistic and associated degrees of freedom.

**ZETA=number**

tunes the sensitivity in forming Type III functions. Any element in the estimable function basis with an absolute value less than *number* is set to 0. The default is 1E–8.
Chapter 49: The HPMIXED Procedure

OUTPUT Statement

```
OUTPUT <OUT=SAS-data-set>
  <keyword<(keyword-options)> <=name>> . . .
  <keyword<(keyword-options)> <=name>> </options> ;
```

The OUTPUT statement creates a data set that contains predicted values and residual diagnostics, computed after fitting the model. By default, all variables in the original data set are included in the output data set.

You can use the ID statement to select a subset of the variables from the input data set to be added to the output data set.

For example, suppose that the data set `Scores` contains the variables `score`, `machine`, and `person`. The following statements fit a model with fixed machine and random person effects and save the predicted and residual values to the data set `igausout`:

```
proc hpmixed data = Scores;
  class machine person score;
  model score = machine;
  random person;
  output out=igausout pred=p resid=r;
run;
```

You can specify the following `options` in the OUTPUT statement before the slash (`/`).

- **OUT**=`SAS data set`
  
  Specifies the name of the output data set. If the OUT= option is omitted, the procedure uses the `DATA`n convention to name the output data set.

- **keyword<(keyword-options)> <=name>**

  Specifies a statistic to include in the output data set and optionally assigns the variable the name `name`. You can use the `keyword-options` to control which type of a particular statistic to compute. The `keyword-options` can take on the following values:

  - **BLUP**
    
    Uses the predictors of the random effects in computing the statistic.

  - **NOBLUP**
    
    Does not use the predictors of the random effects in computing the statistic.

  The default is to compute statistics by using BLUPs. For example, the following two OUTPUT statements are equivalent:

  ```
  output out=out1 pred=predicted lcl=lower;
  output out=out1 pred(blup)=predicted lcl(blup)=lower;
  ```

  If a particular combination of `keyword` and `keyword-options` is not supported, the statistic is not computed and a message is produced in the SAS log.

  A `keyword` can appear multiple times in the OUTPUT statement. Table 49.7 lists the `keywords` and the default names assigned by the HPMIXED procedure if you do not specify a `name`. In this table, `y` denotes the response variable.
### Table 49.7  Keywords for Output Statistics

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Options</th>
<th>Description</th>
<th>Expression</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREDICTED</td>
<td>BLUP</td>
<td>Linear predictor</td>
<td>( \hat{\eta} = x'\hat{\beta} + z'\hat{\gamma} )</td>
<td>Pred</td>
</tr>
<tr>
<td></td>
<td>NOBLUP</td>
<td>Marginal linear predictor</td>
<td>( \hat{\eta}_m = x'\hat{\beta} )</td>
<td>PredPA</td>
</tr>
<tr>
<td>STDERR</td>
<td>BLUP</td>
<td>Standard deviation of linear predictor</td>
<td>( \sqrt{\text{Var}[\hat{\eta} - z'y]} )</td>
<td>StdErr</td>
</tr>
<tr>
<td></td>
<td>NOBLUP</td>
<td>Standard deviation of marginal linear predictor</td>
<td>( \sqrt{\text{Var}[\hat{\eta}_m]} )</td>
<td>StdErrPA</td>
</tr>
<tr>
<td>RESIDUAL</td>
<td>BLUP</td>
<td>Residual</td>
<td>( r = y - \hat{\eta} )</td>
<td>Resid</td>
</tr>
<tr>
<td></td>
<td>NOBLUP</td>
<td>Marginal residual</td>
<td>( r_m = y - \hat{\eta}_m )</td>
<td>ResidPA</td>
</tr>
<tr>
<td>PEARSON</td>
<td>BLUP</td>
<td>Pearson-type residual</td>
<td>( r / \sqrt{\text{Var}[y</td>
<td>y]} )</td>
</tr>
<tr>
<td></td>
<td>NOBLUP</td>
<td>Marginal Pearson-type residual</td>
<td>( r_m / \sqrt{\text{Var}[y]} )</td>
<td>PearsonPA</td>
</tr>
<tr>
<td>STUDENT</td>
<td>BLUP</td>
<td>Studentized residual</td>
<td>( r / \sqrt{\text{Var}[r]} )</td>
<td>Student</td>
</tr>
<tr>
<td></td>
<td>NOBLUP</td>
<td>Studentized marginal residual</td>
<td>( r_m / \sqrt{\text{Var}[r_m]} )</td>
<td>StudentPA</td>
</tr>
<tr>
<td>LCL</td>
<td>BLUP</td>
<td>Lower prediction limit for linear predictor</td>
<td>( \text{LCL} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOBLUP</td>
<td>Lower confidence limit for marginal linear predictor</td>
<td>( \text{LCLPA} )</td>
<td></td>
</tr>
<tr>
<td>UCL</td>
<td>BLUP</td>
<td>Upper prediction limit for linear predictor</td>
<td>( \text{UCL} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOBLUP</td>
<td>Upper confidence limit for marginal linear predictor</td>
<td>( \text{UCLPA} )</td>
<td></td>
</tr>
<tr>
<td>VARIANCE</td>
<td>BLUP</td>
<td>Conditional variance of response variable</td>
<td>( \sqrt{\text{Var}[y</td>
<td>y]} )</td>
</tr>
<tr>
<td></td>
<td>NOBLUP</td>
<td>Marginal variance of response variable</td>
<td>( \sqrt{\text{Var}[y]} )</td>
<td>VariancePA</td>
</tr>
</tbody>
</table>

You can use the following shortcuts to request statistics: PRED for PREDICTED, STD for STDERR, RESID for RESIDUAL, VAR for VARIANCE.

You can specify the following **options** of the OUTPUT statement after the slash (/).

**ALLSTATS**
requests that all statistics are computed. If you do not use a **keyword** to assign a name, the HPMIXED procedure uses the default name.

**ALPHA=number**
determines the coverage probability for two-sided confidence and prediction intervals. The coverage probability is computed as \( 1 - \text{number} \). The value of number must be between 0 and 1 inclusively; the default is 0.05.
NOMISS
requests that records from the input data set be written to the output data only for those observations that were used in the analysis. By default, the HPMIXED procedure produces output statistics for all observations in the input data set.

NOUNIQUE
requests that names not be made unique in the case of naming conflicts. By default, the HPMIXED procedure avoids naming conflicts by assigning a unique name to each output variable. If you specify the NOUNIQUE option, variables with conflicting names are not renamed. In that case, the first variable added to the output data set takes precedence.

NOVAR
requests that variables from the input data set not be added to the output data set. This option ignores ID statement but does not apply to variables listed in a BY statement.

PARMS Statement

PARMS <(value-list)... > < /options> ;

The PARMS statement specifies initial values for the covariance parameters, or it requests a grid search over several values of these parameters. You must specify the values in the order in which they appear in the “Covariance Parameter Estimates” table.

The value-list specification can take any of several forms:

- \( m \) a single value
- \( m_1, m_2, \ldots, m_n \) several values
- \( m \) to \( n \) a sequence where \( m \) equals the starting value, \( n \) equals the ending value, and the increment equals 1
- \( m \) to \( n \) by \( i \) a sequence where \( m \) equals the starting value, \( n \) equals the ending value, and the increment equals \( i \)
- \( m_1, m_2 \) to \( m_3 \) mixed values and sequences

You can use the PARMS statement to input known parameters. Suppose the three variance components are known to be 2, 1, and 3. The SAS statements to fix the variance components at these values are as follows:

```
proc hpmixed noprofile;
  class Family Gender;
  model Height = Gender;
  random Family Family*Gender;
  parms (2) (1) (3) / noiter;
run;
```

The NOPROFILE option in the PROC HPMIXED statement suppresses profiling the residual variance parameter during its calculations, thereby enabling its value to be held at 3 as specified in the PARMS statement.

If you specify more than one set of initial values, PROC HPMIXED performs a grid search of the likelihood surface and uses the best point on the grid for subsequent analysis. Specifying a large number of grid points can result in long computing times. The grid search feature is also useful for exploring the likelihood surface.
The results from the PARMS statement are the values of the parameters on the specified grid (denoted by CovP1–CovPn), the residual variance (possibly estimated) for models with a residual variance parameter, and various functions of the likelihood.

The name of the “Parameter Search” table is ParmSearch.

You can specify the following options in the PARMS statement after a slash (/).

**HOLD=value-list**

**HOLD** specifies which parameter values PROC HPMIXED should hold to equal the specified values. To hold all parameters, you can use the second form without giving the value-list. For example, the following statement constrains the first and third covariance parameters to equal 5 and 2, respectively.

Specifying the HOLD= option implies the NOPROFILE option in the PROC HPMIXED statement:

```
parms (5) (3) (2) (3) / hold=1,3;
```

**LOWERB=value-list**

enables you to specify lower boundary constraints on the covariance parameters. The value-list specification is a list of numbers or missing values (.) separated by commas. You must list the numbers in the order that PROC HPMIXED uses for the covariance parameters, and each number corresponds to the lower boundary constraint. A missing value instructs PROC HPMIXED to use its default constraint, and if you do not specify numbers for all of the covariance parameters, PROC MIXED assumes the remaining ones are missing.

**NOITER** requests that no optimization iterations be performed and that PROC HPMIXED use the best value from the grid search to perform inferences. By default, iterations begin at the best value from the PARMS grid search. This option is ignored when you specify the HOLD= option.

If a residual variance is profiled, the parameter estimates can change from the initial values you provide as the residual variance is recomputed. To prevent an update of the residual variance, combine the NOITER option with the NOPROFILE option in the PROC HPMIXED statements, as in the following program:

```
proc hpmixed noprofile;
  class A B C rep mp sp;
  model y = A | B | C;
  random rep mp sp;
  parms (180) (200) (170) (1000) / noiter;
run;
```

Specifying the NOITER option in the PARMS statement has the same effect as specifying TECHNIQUE=NONE in the NLOPTIONS statement.

Notice that the NOITER option can be useful if you want to obtain the starting values HPMIXED computes. The following statements produce the starting values:
proc hpmixed noprofile;
  class A B;
  model y = A;
  random int / subject=B;
  parms / noiter;
run;

PARMSDATA=SAS-data-set
PDATAL=SAS data set

reads in covariance parameter values from a SAS data set. The data set should contain the numerical variable ESTIMATE or the numerical variables Covp1–Covpq, where q denotes the number of covariance parameters.

If the PARMSDATA= data set contains multiple sets of covariance parameters, the HPMIXED procedure evaluates the initial objective function for each set and commences the optimization step by using the set with the lowest function value as the starting values. For example, the following SAS statements request that the objective function be evaluated for three sets of initial values:

    data data_covp;
      input covp1-covp4;
      datalines;
      180 200 170 1000
      170 190 160 900
      160 180 150 800
    ;
    proc hpmixed;
      class A B C rep;
      model yield = A;
      random rep B C;
      parms / pdata=data_covp;
    run;

Each set comprises four covariance parameters.

The order of the observations in a data set with the numerical variable Estimate corresponds to the order of the covariance parameters in the “Covariance Parameter Estimates” table.

The PARMSDATA= data set must contain at least one set of covariance parameters with no missing values.

If the HPMIXED procedure is processing the input data set in BY groups, you can add the BY variables to the PARMSDATA= data set. If this data set is sorted by the BY variables, the HPMIXED procedure matches the covariance parameter values to the current BY group. If the PARMSDATA= data set does not contain all BY variables, the data set is processed in its entirety for every BY group and a message is written to the log. This enables you to provide a single set of starting values across BY groups, as in the following statements:

    data data_covp;
      input covp1-covp4;
      datalines;
The same set of starting values is used for each value of the `year` variable.

**UPPERB=value-list**

enables you to specify upper boundary constraints on the covariance parameters. The `value-list` specification is a list of numbers or missing values (.) separated by commas. You must list the numbers in the order that PROC HPMIXED uses for the covariance parameters, and each number corresponds to the upper boundary constraint. A missing value instructs PROC HPMIXED to use its default constraint, and if you do not specify numbers for all of the covariance parameters, PROC HPMIXED assumes that the remaining ones are missing.

**RANDOM Statement**

```
RANDOM random-effects < / options > ;
```

The RANDOM statement defines the random effects in the mixed model. It can be used to specify traditional variance component models (as in the VARCOMP procedure) and to specify random coefficients. The random effects can be classification or continuous. Multiple RANDOM statements are possible. Random effects specified in a RANDOM statement could be correlated with each other for certain types of covariance structures (see the **TYPE=** option on page 3851). It is, however, assumed that random effects specified using different RANDOM statements are not correlated.

Using notation from the section “Model Assumptions” on page 3857, the purpose of the RANDOM statement is to define the \( Z \) matrix of the mixed model, the random effects in the \( y \) vector, and the structure of \( G \). The \( Z \) matrix is constructed exactly like the \( X \) matrix for the fixed effects, and the \( G \) matrix is constructed to correspond to the effects constituting \( Z \). The structure of \( G \) is defined by using the **TYPE=** option described on page 3851.

You can specify INTERCEPT (or INT) as a random effect. PROC HPMIXED does not include the intercept in the RANDOM statement by default, as it does in the **MODEL** statement.

You can specify the following **options** in the RANDOM statement after a slash (/).

**ALPHA=number**

requests that a \( t \)-type confidence interval with confidence level \( 1 - \text{number} \) be constructed for the predictors of random effects in this statement. The value of \text{number} must be between 0 and 1 exclusively; the default is 0.05. Specifying the **ALPHA=** option implies the **CL** option.
requests that \( t \)-type confidence limits be constructed for each of the predictors of random effects in this statement. The confidence level is 0.95 by default; this can be changed with the ALPHA= option. The CL option implies the SOLUTION option.

defines an effect specifying heterogeneity in the covariance structure of \( G \). All observations having the same level of the group effect have the same covariance parameters. Each new level of the group effect produces a new set of covariance parameters with the same structure as the original group. You should exercise caution in defining the group effect, because strange covariance patterns can result from its misuse. Also, the group effect can greatly increase the number of estimated covariance parameters, which can adversely affect the optimization process.

Continuous variables are permitted as arguments to the GROUP= option. PROC HPMIXED does not sort by the values of the continuous variable; rather, it considers the data to be from a new group whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of groups and also prevents the production of a large “Class Levels Information” table.

eliminates the columns in \( Z \) corresponding to missing levels of random effects involving CLASS variables. By default, these columns are included in \( Z \). It is sufficient to specify the NOFULLZ option in any RANDOM statement.

requests that the solution for the random-effects parameters be produced. Using notation from the section “Model Assumptions” on page 3857, these estimates are the empirical best linear unbiased predictors (BLUPs) \( \hat{\beta} = \hat{G}Z\hat{V}^{-1}(y - X\hat{\beta}) \). They can be useful for comparing the random effects from different experimental units and can also be treated as residuals in performing diagnostics for your mixed model.

The numbers displayed in the SE Pred column of the “Solution for Random Effects” table are not the standard errors of the \( \hat{\beta} \) displayed in the Estimate column; rather, they are the standard errors of predictions \( \hat{y}_i - y_i \), where \( \hat{y}_i \) is the \( i \)th BLUP and \( y_i \) is the \( i \)th random-effect parameter.

identifies the subjects in your mixed model. Complete independence is assumed across subjects; thus, for the RANDOM statement, the SUBJECT= option produces a block-diagonal structure in \( G \) with identical blocks. The \( Z \) matrix is modified to accommodate this block-diagonality. In fact, specifying a subject effect is equivalent to nesting all other effects in the RANDOM statement within the subject effect.

Continuous variables are permitted as arguments to the SUBJECT= option. PROC HPMIXED does not sort by the values of the continuous variable; rather, it considers the data to be from a new subject whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of subjects and also prevents the production of a large “Class Levels Information” table.
**TYPE=**covariance-structure

specifies the structure of the covariance matrix G for random effects. The default structure is VC.

If you want different covariance structures in different parts of G, you must use multiple RANDOM statements with different TYPE= options.

Valid values for covariance-structure are listed in Table 49.8. Examples are shown in Table 49.9.

### Table 49.8 Covariance Structures

<table>
<thead>
<tr>
<th>Structure</th>
<th>Description</th>
<th>Parameters</th>
<th>(i, j) element</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR(1)</td>
<td>Autoregressive(1)</td>
<td>2</td>
<td>$\sigma^2 \rho^{i-j}$</td>
</tr>
<tr>
<td>CHOL</td>
<td>Cholesky root</td>
<td>$t(t+1)/2$</td>
<td>$l_{ij}$</td>
</tr>
<tr>
<td>CS</td>
<td>Compound symmetry (CS)</td>
<td>2</td>
<td>$\sigma_1 + \sigma^2 1(i = j)$</td>
</tr>
<tr>
<td>CSH</td>
<td>Heterogeneous CS</td>
<td>$t+1$</td>
<td>$\sigma_i \sigma_j [\rho 1(i \neq j) + 1(i = j)]$</td>
</tr>
<tr>
<td>UC</td>
<td>Uniform correlation (UC)</td>
<td>2</td>
<td>$\sigma^2 [\rho 1(i \neq j) + 1(i = j)]$</td>
</tr>
<tr>
<td>UCH</td>
<td>Heterogeneous UC</td>
<td>$t+1$</td>
<td>$\sigma_i \sigma_j [\rho 1(i \neq j) + 1(i = j)]$</td>
</tr>
<tr>
<td>UN</td>
<td>Unstructured</td>
<td>$t(t+1)/2$</td>
<td>$\sigma_{ij}$</td>
</tr>
<tr>
<td>VC</td>
<td>Variance components</td>
<td>q</td>
<td>$\sigma_k^2 1(i = j)$</td>
</tr>
</tbody>
</table>

In Table 49.8, $t$ is the overall dimension of the covariance matrix, and $1(A)$ equals 1 when A is true and 0 otherwise. For example, $1(i = j)$ equals 1 when $i = j$ and equals 0 otherwise. TYPE=UCH is the same as TYPE=CSH.

### Table 49.9 Covariance Structure Examples

<table>
<thead>
<tr>
<th>Description</th>
<th>Structure</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>First-order</td>
<td>AR(1)</td>
<td>$\sigma^2 \begin{bmatrix} 1 &amp; \rho &amp; \rho^2 &amp; \rho^3 \ \rho &amp; 1 &amp; \rho &amp; \rho^2 \ \rho^2 &amp; \rho &amp; 1 &amp; \rho \ \rho^3 &amp; \rho^2 &amp; \rho &amp; 1 \end{bmatrix}$</td>
</tr>
<tr>
<td></td>
<td>CHOL</td>
<td>$\begin{bmatrix} l_{11} &amp; 0 &amp; 0 &amp; 0 \ l_{21} &amp; l_{22} &amp; 0 &amp; 0 \ l_{31} &amp; l_{32} &amp; l_{33} &amp; 0 \ l_{41} &amp; l_{42} &amp; l_{43} &amp; l_{44} \end{bmatrix}$</td>
</tr>
<tr>
<td></td>
<td>CS</td>
<td>$\begin{bmatrix} \sigma^2 + \sigma_1 &amp; \sigma_1 &amp; \sigma_1 &amp; \sigma_1 \ \sigma_1 &amp; \sigma^2 + \sigma_1 &amp; \sigma_1 &amp; \sigma_1 \ \sigma_1 &amp; \sigma_1 &amp; \sigma^2 + \sigma_1 &amp; \sigma_1 \ \sigma_1 &amp; \sigma_1 &amp; \sigma_1 &amp; \sigma^2 + \sigma_1 \end{bmatrix}$</td>
</tr>
<tr>
<td>Description</td>
<td>Structure</td>
<td>Example</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| Uniform correlation  | UC        | \[
\begin{bmatrix}
1 & \rho & \rho & \rho \\
\rho & 1 & \rho & \rho \\
\rho & \rho & 1 & \rho \\
\rho & \rho & \rho & 1 \\
\end{bmatrix}
\]
| Heterogeneous        | UCH       | \[
\begin{bmatrix}
\sigma_1^2 & \sigma_1 \sigma_2 \rho & \sigma_1 \sigma_3 \rho & \sigma_1 \sigma_4 \rho \\
\sigma_2 \sigma_1 \rho & \sigma_2^2 & \sigma_2 \sigma_3 \rho & \sigma_2 \sigma_4 \rho \\
\sigma_3 \sigma_1 \rho & \sigma_3 \sigma_2 \rho & \sigma_3^2 & \sigma_3 \sigma_4 \rho \\
\sigma_4 \sigma_1 \rho & \sigma_4 \sigma_2 \rho & \sigma_4 \sigma_3 \rho & \sigma_4^2 \\
\end{bmatrix}
\]
| Unstructured         | UN        | \[
\begin{bmatrix}
\sigma_1^2 & \sigma_{21} & \sigma_{31} & \sigma_{41} \\
\sigma_{21} & \sigma_2^2 & \sigma_{32} & \sigma_{42} \\
\sigma_{31} & \sigma_{32} & \sigma_3^2 & \sigma_{34} \\
\sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_4^2 \\
\end{bmatrix}
\]
| Variance components  | VC (default) | \[
\begin{bmatrix}
\sigma_A^2 & 0 & 0 & 0 \\
0 & \sigma_A^2 & 0 & 0 \\
0 & 0 & \sigma_B^2 & 0 \\
0 & 0 & 0 & \sigma_B^2 \\
\end{bmatrix}
\]

The variances and covariances in the formulas that follow in the TYPE= descriptions are expressed in terms of generic random variables \( \xi_i \) and \( \xi_j \). They represent random effects for which the \( \mathbf{G} \) matrices are constructed.

The following list provides some further information about these covariance-structures:

**AR(1)** specifies a first-order autoregressive structure,

\[
\text{Cov} \left[ \xi_i, \xi_j \right] = \sigma^2 \rho^{|i-j|}
\]

The values \( i \) and \( j \) are derived for the \( i \)th and \( j \)th observations, respectively. For example, in the following statements the values correspond to the class levels for the time effect of the \( i \)th and \( j \)th observation within a particular subject:

```r
proc hpmixed;
  class time patient;
  model y = x x*x;
  random time / sub=patient type=ar(1);
run;
```

PROC HPMIXED imposes the constraint \( |\rho| < 1 \) for stationarity.

**CHOL** specifies an unstructured variance-covariance matrix parameterized through its Cholesky root. All diagonal values are constrained to be positive. This parameterization guarantees a positive definite covariance matrix. For example, a \( 2 \times 2 \) unstructured covariance matrix can be written as

\[
\text{Var} \left[ \mathbf{\xi} \right] = \begin{bmatrix}
\sigma_1^2 & \sigma_{21} \\
\sigma_{21} & \sigma_2^2 \\
\end{bmatrix}
\]
Without imposing constraints on the three parameters, there is no guarantee that the estimated variance matrix is positive definite. Even if $\sigma_1^2$ and $\sigma_2^2$ are nonzero, a large value for $\sigma_{21}$ can lead to a negative eigenvalue of $\text{Var}[\xi]$. The Cholesky root of a positive definite matrix $A$ is a lower triangular matrix $L$ such that $LL' = A$. The Cholesky root of the above $2 \times 2$ matrix can be written as

$$L = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix}$$

The elements of the unstructured variance matrix are then simply $\sigma_1^2 = l_{11}^2$, $\sigma_{21} = l_{21}l_{11}$, and $\sigma_2^2 = l_{21}^2 + l_{22}^2$. Similar operations yield the generalization to covariance matrices of higher orders.

For example, the following statements model the covariance matrix of each subject as an unstructured matrix:

```plaintext
proc hpmixed;
  class sub;
  model y = x;
  random time / sub=patient type=chol;
run;
```

The HPMIXED procedure constrains the diagonal elements of the Cholesky root to be positive. This guarantees that the structure is positive definite.

**CS**

specifies the compound-symmetry structure, which has constant variance and constant covariance

$$\text{Cov}[\xi_i, \xi_j] = \begin{cases} \sigma^2 + \sigma_1 & i = j \\ \sigma_1 & i \neq j \end{cases}$$

Under compound-symmetry, the $G$ matrix is of form $\sigma^2 I + \sigma_1 J$. The variance parameter $\sigma^2$ is constrained to be positive, and the covariance parameter $\sigma_1$ is constrained to be greater than $-\sigma^2/t$ where $t$ is the dimension of the structure. This guarantees the structure is positive definite. The compound-symmetry structure arises naturally with nested random effects, such as when a subsampling error is nested within an experimental error.

**CSH**

specifies the heterogeneous compound-symmetry structure. This structure has a different variance parameter for each diagonal element, and it uses the square roots of these parameters in the off-diagonal entries. In Table 49.8, $\sigma_i^2$ is the $i$th variance parameter that satisfies $\sigma_i^2 > 0$, and $\rho$ is the correlation parameter that satisfies $\rho > -1/(t-1)$, where $t$ is the dimension of the structure. This guarantees that the structure is positive definite.

**UC**

specifies the uniform correlation structure, which has constant variance and constant correlation

$$\text{Cov}[\xi_i, \xi_j] = \begin{cases} \sigma^2 & i = j \\ \sigma^2 \rho & i \neq j \end{cases}$$

Under uniform correlation, the $G$ matrix is of form $\sigma^2[(1-\rho)I + \rho J]$. The variance $\sigma^2$ is constrained to be positive, and the correlation $\rho$ is constrained to be greater
than \(-1/(t-1)\), where \(t\) is the dimension of the structure. This guarantees the structure is positive definite. This structure is equivalent to the compound-symmetry structure with a better numerical property in terms of optimization.

The uniform correlation structure arises frequently in agriculture and animal sciences.

**UCH** specifies the heterogeneous uniform correlation structure. This structure has a different variance parameter for each diagonal element, and it uses the square roots of these parameters in the off-diagonal entries. In Table 49.8, \(\sigma_i^2\) is the \(i\)th variance parameter that satisfies \(\sigma_i^2 > 0\), and \(\rho\) is the correlation parameter that satisfies \(\rho > -1/(t-1)\), where \(t\) is the dimension of the structure. This guarantees that the structure is positive definite.

**UN** specifies a completely general (unstructured) covariance matrix parameterized directly in terms of variances and covariances. The variances are constrained to be positive, and the covariances are unconstrained. In addition, this structure is internally constrained to be positive definite.

**VC** specifies standard variance components and is the default structure for the RANDOM and REPEATED statements. In the RANDOM statement, a distinct variance component is assigned to each effect. In the REPEATED statement, this structure is usually used only with the GROUP= option to specify a heterogeneous variance model.

---

**REPEATED Statement**

```
REPEATED repeated-effect < / options > ;
```

The **REPEATED** statement defines the repeated effect and the residual covariance structure in the mixed model. The residual variance-covariance matrix is denoted as \(R\). The **repeated-effect** is required and consists entirely of classification variables. The levels of the **repeated-effect** must be different for each observation within a subject in order to avoid the singular \(R\) matrix. The **SUBJECT=** option is required. The data set must be grouped by subject effect.

Table 49.10 summarizes the **options** available in the **REPEATED** statement.

<table>
<thead>
<tr>
<th><strong>Option</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Construction of Covariance Structure</strong></td>
<td></td>
</tr>
<tr>
<td>GROUP=</td>
<td>Defines an effect that specifies heterogeneity in the residual covariance structure</td>
</tr>
<tr>
<td>SUBJECT=</td>
<td>Identifies the subjects in the residual covariance structure</td>
</tr>
<tr>
<td>TYPE=</td>
<td>Specifies the residual covariance structure (the default is VC)</td>
</tr>
<tr>
<td><strong>Statistical Output</strong></td>
<td></td>
</tr>
<tr>
<td>R=</td>
<td>Displays blocks of the estimated (R) matrix</td>
</tr>
<tr>
<td>RC=</td>
<td>Display the Cholesky root (lower) of blocks of the estimated (R) matrix</td>
</tr>
</tbody>
</table>
Table 49.10  continued

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCI=</td>
<td>Displays the inverse Cholesky root (lower) of blocks of the estimated $R$ matrix</td>
</tr>
<tr>
<td>RCORR=</td>
<td>Displays the correlation matrix that corresponds to blocks of the estimated $R$ matrix</td>
</tr>
<tr>
<td>RI=</td>
<td>Displays the inverse of blocks of the estimated $R$ matrix</td>
</tr>
</tbody>
</table>

You can specify the following options in the REPEATED statement after a slash (/).

GROUP=effect

defines an effect that specifies heterogeneity in the residual covariance structure. All observations that have the same level of the GROUP effect have the same covariance parameters. Each new level of the GROUP effect produces a new set of covariance parameters with the same structure as the original group. You should exercise caution in defining the GROUP effect, because strange covariance patterns can result with its misuse. Also, the GROUP effect can greatly increase the number of estimated covariance parameters, which can adversely affect the optimization process.

Continuous variables are permitted as arguments to the GROUP= option. PROC HPMIXED does not sort by the values of the continuous variable; rather, it considers the data to be from a new subject or group whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of subjects or groups and also prevents the production of a large “Class Level Information” table.

$R<value-list>$

requests that blocks of the estimated $R$ matrix be displayed. The first block determined by the SUBJECT= effect is the default displayed block.

The value-list indicates the subjects for which blocks of $R$ are to be displayed. For example, the following statement displays block matrices for the first, third, and fifth persons:

repeated time / type=un subject=person r=1,3,5;

See the PARMS statement for the possible forms of value-list.

RC<value-list>

displays the Cholesky root of blocks of the estimated $R$ matrix. The value-list specification is the same as for the R= option.

RCI<value-list>

displays the inverse Cholesky root of blocks of the estimated $R$ matrix. The value-list specification is the same as for the R= option.

RCORR<value-list>

displays the correlation matrix that corresponds to blocks of the estimated $R$ matrix. The value-list specification is the same as for the R= option.
RI<value-list>
produces the inverse of blocks of the estimated R matrix. The value-list specification is the same as for the R= option.

SUBJECT=effect
SUB=effect
identifies the subjects in your mixed model. Complete independence is assumed across subjects; therefore, the SUBJECT= option produces a block-diagonal structure in R with identical blocks. The SUBJECT= option is required. The data set must be grouped by SUBJECT= effect. When the SUBJECT= effect consists entirely of classification variables, the blocks of R correspond to observations that share the same level of that effect. These blocks are sorted according to this effect as well.

Continuous variables are permitted as arguments to the SUBJECT= option. PROC HPMIXED does not sort by the values of the continuous variable; rather, it considers the data to be from a new subject or group whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of subjects or groups and also prevents the production of a large “Class Level Information” table.

If you want to model nonzero covariance among all of the observations in your data, specify SUBJECT=INTERCEPT to treat the data as if they are all from one subject. However, be aware that in this case PROC HPMIXED manipulates an R matrix with dimensions equal to the number of observations.

TYPE=covariance-structure
specifies the structure of the residual variance-covariance matrix R. The SUBJECT= option defines the blocks of R, and the TYPE= option specifies the structure of these blocks. PROC HPMIXED supports the following structures: TYPE=AR(1), TYPE=CHOL, TYPE=UN, and TYPE=VC. The default structure is VC. See the description in the section “RANDOM Statement” on page 3849 for more information about these covariance structure types.

TEST Statement

TEST fixed-effects < / options > ;

The TEST statement performs a hypothesis test on the fixed effects. You can specify multiple effects in one TEST statement or in multiple TEST statements, and all TEST statements must appear after the MODEL statement.

You can specify the following options in the TEST statement after a slash (/).

HTYPE=value-list
indicates the type of hypothesis test to perform on the specified effects. Valid entries for values in the value-list are 3, corresponding to a Type III test. The default value is 3. The ODS table name is “Tests3” for the Type III test.

E
requests that matrix coefficients associated with test types be displayed for specified effects.
requests that Type III matrix coefficients be displayed if a Type III test is performed.

CHISQ
requests that \( \chi^2 \) tests be performed in addition to any \( F \) tests. A \( \chi^2 \) statistic equals its corresponding \( F \) statistic times the associate numerator degree of freedom, and this same degree of freedom is used to compute the \( p \)-value for the \( \chi^2 \) test. This \( p \)-value will always be less than that for the \( F \) test, because it effectively corresponds to an \( F \) test with infinite denominator degrees of freedom.

WEIGHT Statement

```
WEIGHT variable;
```

The WEIGHT statement replaces \( R \) with \( W^{-1/2}RW^{-1/2} \), where \( W \) is a diagonal matrix containing the weights. Observations with nonpositive or missing weights are not included in the resulting PROC HPMIXED analysis. If a WEIGHT statement is not included, all observations used in the analysis are assigned a weight of 1.

If a computation in PROC MIXED involves \( R \), then the WEIGHT statement replaces \( R \) with \( W^{-1/2}RW^{-1/2} \). For example, the covariance matrix \( V \) for the observations usually have the form \( V = ZGZ' + R \); therefore, with the WEIGHT statement, this becomes \( V = ZGZ' + W^{-1/2}RW^{-1/2} \).

Details: HPMIXED Procedure

Model Assumptions

The following sections provide an overview of the approach used by the HPMIXED procedure for likelihood-based analysis of linear mixed models with sparse matrix technique. Additional theory and examples are provided in Littell et al. (1996); Verbeke and Molenberghs (1997, 2000); Brown and Prescott (1999).

The HPMIXED procedure fits models generally of the form

\[
y = X\beta + Z\gamma + \epsilon
\]

Models of this form contain both fixed-effects parameters, \( \beta \), and random-effects parameters, \( \gamma \); hence, they are called *mixed models*. See Henderson (1990) and Searle, Casella, and McCulloch (1992) for historical developments of the mixed model. Note that the matrix \( Z \) can contain either continuous or dummy variables, just like \( X \).

So far this is the same general form of model fit by the MIXED procedure. The difference between the models handled by the two procedures lies in the assumptions about the distributions of \( \gamma \) and \( \epsilon \). For both procedures a key assumption is that \( \gamma \) and \( \epsilon \) are normally distributed with

\[
\begin{bmatrix}
E[y] \\
E[\epsilon]
\end{bmatrix} = 
\begin{bmatrix}
0 \\
0
\end{bmatrix}
\]

\[
\text{Var}\left[\begin{bmatrix}
y \\
\epsilon
\end{bmatrix}\right] = 
\begin{bmatrix}
G & 0 \\
0 & R
\end{bmatrix}
\]
The two procedures differ in their assumptions about the variance matrices $G$ and $R$ for $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$, respectively. The MIXED procedure allows a variety of different structures for both $G$ and $R$; while in HPMIXED procedure, $R$ is always assumed to be of the form $R = I \sigma^2$, and the structures available for modeling $G$ are only a small subset of the structures offered by the MIXED procedure.

Estimates of fixed effects and predictions for random effects are obtained by solving the so-called mixed model equations:

$$
\begin{bmatrix}
X'X/\sigma^2 & X'Z/\sigma^2 \\
Z'X/\sigma^2 & Z'Z/\sigma^2 + G^{-1}
\end{bmatrix}
\begin{bmatrix}
\widehat{\beta} \\
\widehat{\gamma}
\end{bmatrix}
= 
\begin{bmatrix}
X'y/\sigma^2 \\
Z'y/\sigma^2
\end{bmatrix}
$$

Let $C$ denote the coefficient matrix of the mixed model equations:

$$
C = 
\begin{bmatrix}
X'X/\sigma^2 & X'Z/\sigma^2 \\
Z'X/\sigma^2 & Z'Z/\sigma^2 + G^{-1}
\end{bmatrix}
$$

Under the assumptions given previously for the moments of $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$, the variance of $y$ is $V = ZGZ' + I \sigma^2$. You can model $V$ by setting up the random-effects design matrix $Z$ and by specifying covariance structures for $G$. Let $\theta$ be a vector of all unknown parameters in $G$. Then the general form of the restricted likelihood function for the mixed models that the HPMIXED procedure can fit is

$$
L(\theta, \sigma^2) = -2 \log l = (n - p) \log(2\pi) + \log |C| + \log |G| + n \log(\sigma^2) + y'Py
$$

where

$$
P = V^{-1} - V^{-1}X(X'V^{-1}X)^{-1}X'V^{-1}
$$

and $p$ is the rank of $X$. The HPMIXED procedure minimizes $L(\theta, \sigma^2)$ over all unknown parameters in $\theta$ and $\sigma^2$ by using nonlinear optimization algorithms.

### Computing and Maximizing the Likelihood

In computing the restricted likelihood function given previously, the determinants of the matrices $C$ and $G$ can be obtained effectively by using Cholesky decomposition. The quadratic term $y'Py$ can be expressed in terms of solutions of mixed model equations as follows:

$$
y'Py = \frac{1}{\sigma^2} \left( y'y - [\widehat{\beta}', \widehat{\gamma}'] \begin{bmatrix} X'y \\ Z'y \end{bmatrix} \right)
$$

By default, the HPMIXED procedure profiles out the residual variance $\sigma^2$ from the parameter vector $\theta$. Let $\theta^*$ be the new parameter vector such that $\theta^*_i = \theta_i / \sigma^2$. The profiled objective function becomes

$$
L(\theta^*, \sigma^2) = (n - p) \log(2\pi) + \log |C^*| + \log |G^*| - (r_C - r_G - n) \log(\sigma^2) + (n - p)
$$

where $C^* = C\sigma^2$ and $G^* = G\sigma^2$ are the profiled versions of $C$ and $G$, $r_C$ and $r_G$ are the ranks of $C$ and $G$. Minimizing analytically for $\sigma^2$ yields

$$
\hat{\sigma}^2 = \frac{1}{n - p} \left( y'y - [\widehat{\beta}', \widehat{\gamma}'] \begin{bmatrix} X'y \\ Z'y \end{bmatrix} \right)
$$
Optimizing the likelihood calls for derivatives with respect to the parameters. The first and second derivatives of the log-likelihood function \( L \) with respect to scalar variance components \( \theta_i \) and \( \theta_j \) are

\[
\frac{\partial L}{\partial \theta_i} = \text{tr} \left( \frac{\partial V}{\partial \theta_i} P \right) - y' P \frac{\partial V}{\partial \theta_i} P y
\]

and

\[
\frac{\partial^2 L}{\partial \theta_i \partial \theta_j} = -\text{tr} \left( \frac{\partial V}{\partial \theta_i} P \frac{\partial V}{\partial \theta_j} P \right) + 2 y' P \frac{\partial V}{\partial \theta_i} P \frac{\partial V}{\partial \theta_j} P y
\]

The default quasi-Newton method of optimization for the HPMIXED procedure requires only first derivatives of the log likelihood, and these are readily derived by solving the mixed model equations. For example, when \( G = I \sigma_a \), the first derivative of the log likelihood with respect to the parameter \( \sigma_a^2 \) can be computed as follows:

\[
\frac{\partial L}{\partial \sigma_a^2} = \frac{q}{\sigma_a^2} - \frac{\text{tr}(C^{aa})}{\sigma_a^4} - \frac{\hat{y}' \hat{y}}{\sigma_a^4}
\]

where \( q \) is the size of \( y \) vector and \( C^{aa} \) is the part of the \( g \)-inverse of the mixed model equation coefficient matrix \( C \) corresponding to the random effect \( y \).

The second derivative of the log likelihood needs to be computed only if you specify certain nondefault optimization techniques in the NLOPTI0NS statement, namely TECH=NEWRAP, TECH=NRRIDG, or TECH=TRUEREG; see “NLOPTI0NS Statement” on page 488 in Chapter 19, “Shared Concepts and Topics,” for more information about optimization techniques. For these second-derivative-based optimization techniques, the HPMIXED procedure does not actually use the true second derivative matrix, or observed information matrix, as defined earlier. Instead, it uses an alternative matrix that is more efficient to compute for large problems and that can be more stable. This alternative is called the average information matrix, and it is defined as follows. The expected value of the second derivative is

\[
E \left( \frac{\partial^2 L}{\partial \theta_i \partial \theta_j} \right) = \text{tr} \left( \frac{\partial V}{\partial \theta_i} P \frac{\partial V}{\partial \theta_j} P \right)
\]

It is this trace that is computationally inefficient to evaluate. But if you average the expected information matrix defined by this formula with the observed information matrix defined by the preceding formula for the true second derivative, then the trace term cancels, leaving just a quadratic expression in \( y \). This quadratic expression defines the average information (Johnson and Thompson 1995) with respect to \( \theta_i \) and \( \theta_j \):

\[
AI(\theta_i, \theta_j) = y' P \frac{\partial V}{\partial \theta_i} P \frac{\partial V}{\partial \theta_j} P y
\]

**Computing Starting Values by EM-REML**

The EM-REML algorithm (Dempster, Laird, and Rubin 1977) iteratively alternates between an expectation step and a maximization step to maximize the restricted log likelihood. The algorithm is based on augmenting the observed data \( y \) with the unobservable random effects \( \hat{y} \), leading to a simplified form for the log likelihood. For example, if \( G = I \sigma_a^2 \) then given the realized values \( y \) of the unobservable random effects \( \hat{y} \), the REML estimate of \( \sigma_a^2 \) satisfies

\[
\hat{\sigma}_a^2 = \frac{\hat{y}' \hat{y}}{q - \sigma^2 / \sigma_a^2 \text{tr}(C^{aa})}
\]
This corresponds to the maximization step of EM-REML. However, the true realized values \( \tilde{y} \) are unknown in practice. The expectation step of EM-REML replaces them with the conditional expected values \( \tilde{y} \) of the random effects, given the observed data \( y \) and initial values for the parameters. The new estimate of \( \sigma^2_A \) is used in turn to recalculate the conditional expected values, and the iteration is repeated until convergence.

It is well known that EM-REML is generally more robust against a poor choice of starting values than general nonlinear optimization methods such as Newton-Raphson, though it tends to converge slowly as it approaches the optimum. The Newton-Raphson method, on the other hand, converges much faster when it has a good set of starting values. The HPMIXED procedure, thus, employs a scheme that uses EM-REML initially in order to get good starting values, and after a few iterations, when the decrease in log likelihood has significantly slowed down, switching to a more general nonlinear optimization technique (by default, quasi-Newton).

---

**Sparse Matrix Techniques**

A key component of the HPMIXED procedure is the use of sparse matrix techniques for computing and optimizing the likelihood expression given in the section “Model Assumptions” on page 3857. There are two aspects to sparse matrix techniques, namely, sparse matrix storage and sparse matrix computations. Typically, computer programs represent an \( N \times M \) matrix in a dense form as an array of size \( NM \), making row-wise and column-wise arithmetic operations particularly efficient to compute. However, if many of these \( NM \) numbers are zeros, then correspondingly many of these operations are unnecessary or trivial. Sparse matrix techniques exploit this fact by representing a matrix not as a complete array, but as a set of nonzero elements and their location (row and column) within the matrix. Sparse matrix techniques are more efficient if there are enough zero-element operations in the dense form to make the extra time required to find and operate on matrix elements in the sparse form worthwhile.

The following discussion illustrates sparse techniques. Let the symmetric matrix \( C \) be the matrix of mixed model equations of size \( 5 \times 5 \).

\[
C = \begin{bmatrix}
8.0 & 0 & 0 & 2.0 & 0 \\
0 & 4.0 & 3.0 & 0 & 0 \\
0 & 3.0 & 5.0 & 0 & 0 \\
2.0 & 0 & 0 & 7.0 & 0 \\
0 & 0 & 0 & 0 & 9.0 \\
\end{bmatrix}
\]

There are 15 elements in the upper triangle of \( C \), though eight of them are zeros. The row and column indices and the values of seven nonzero elements are listed as follows:

\[
\begin{array}{c|cccccccc}
& 1 & 1 & 2 & 2 & 3 & 4 & 5 \\
\hline
1 & 4 & 2 & 3 & 3 & 4 & 5 \\
C_{ij} & 8.0 & 2.0 & 4.0 & 3.0 & 5.0 & 7.0 & 9.0 \\
\end{array}
\]

The most elegant scheme to store these seven elements is to store them in a hash table with row and column indices as a hash key. However, this scheme is not efficient as the number of non-zero elements gets very large. The classical and widely used scheme, and the one the HPMIXED procedure employs, is the \((i_c, j_c, c)\) format, in which the nonzero elements are stored contiguously row by row in the vector \( c \). To identify the individual nonzero elements in each row, you need to know the column index of an element. These column indices are stored in the vector \( j_c \); that is, if \( c(k) = C_{ij} \), then \( j_c(k) = j \). To identify the individual rows, you need to know where each row starts and ends. These row starting positions are stored in the vector \( i_c \). For
instance, if \( C_{ij} \) is the first nonzero element in the row \( i \) and \( c(k) = C_{ij} \), then \( i_c(i) = k \). The row \( i \) ending position is one less than \( i_c(i + 1) \). Thus, the number of nonzero elements in the row \( i \) is \( i_c(i + 1) - i_c(i) \), these elements in the row \( i \) are stored consecutively starting from the position \( k_i = i_c(i) \)

\[ c(k_i), c(k_i + 1), c(k_i + 2), ..., c(k_i+1 - 1) \]

and the corresponding columns indices are stored consecutively in

\[ j_c(k_i), j_c(k_i + 1), j_c(k_i + 2), ..., j_c(k_i+1 - 1) \]

For example, the seven nonzero elements in matrix \( C \) are stored in \((i_c, j_c, c)\) format as

| \( i_c \) | 1 | 3 | 5 | 6 | 7 | 8 |
| \( j_c \) | 1 | 4 | 2 | 3 | 3 | 4 | 5 |
| \( c \)   | 8.0 | 2.0 | 4.0 | 3.0 | 5.0 | 7.0 | 9.0 |

Note that since matrices are stored row by row in the \((i_c, j_c, c)\) format, row-wise operations can be performed efficiently but it is inefficient to retrieve elements column-wise. Thus, this representation will be inefficient for matrix computations requiring column-wise operations. Fortunately, the likelihood calculations for mixed models can usually avoid column-wise operations.

In mixed models, sparse matrices typically arise from a large number of levels for fixed effects and/or random effects. If a linear model contains one or more large CLASS effects, then the mixed model equations are usually very sparse. Storing zeros in mixed model equations not only requires significantly more memory but also results in longer execution time and larger rounding error. As an illustration, the example in the “Getting Started: HPMIXED Procedure” on page 3822 has 3506 mixed model equations. Storing just the upper triangle of these equations in a dense form requires \( \frac{3506^2}{2} = 6,147,771 \) elements. However, there are only 60,944 nonzero elements—less than 1% of what dense storage requires.

Note that as the density of the mixed model equations increases, the advantage of sparse matrix techniques decreases. For instance, a classical regression model typically has a dense coefficient matrix, though the dimension of the matrix is relatively small.

The HPMIXED procedure employs sparse matrix techniques to store the nonzero elements in the mixed model equations and to compute a sparse Cholesky decomposition of these equations. A reordering of the mixed model equations is required in order to keep the minimum memory consumption during the factorization. This reordering process results in a different \( g \)-inverse from what is produced by most other SAS/STAT procedures, for which the \( g \)-inverse is defined by sequential sweeping in the order defined by the model. If mixed model equations are singular, this different \( g \)-inverse produces a different solution of mixed model equations. However, estimable functions and tests based on them are invariant to the choice of \( g \)-inverse, and are thus the same for the HPMIXED procedure as for other procedures.

**Hypothesis Tests for Fixed Effects**

Unlike most other SAS/STAT procedures for analyzing general linear models, the HPMIXED procedure does not by default provide \( F \) tests for the fixed effects. This is because, for the large mixed model problems that the HPMIXED procedure is designed to address, such tests are often computationally prohibitive to compute.
The computation of Type III tests first constructs the Hermite matrix of the mixed model coefficient matrix $C$ and then forms the $L$ coefficient matrix to obtain the $F$ value as follows:

$$ F = \frac{\left[ \hat{\beta} \right]' L'(L\hat{C}^{-1}L')^{-1}L \left[ \hat{\beta} \right]}{r} $$

where $r = \text{rank}(L\hat{C}^{-1}L')$. The coefficient matrix $L$ corresponding to fixed effects with many levels can be very large and dense, making them very difficult to work with. At the same time, Type III tests for effects with many levels are relatively unlikely to be statistically useful.

For this reason, you must use the TEST statement in PROC HPMDIXED to specifically ask for Type III tests for any effects for which you want to compute them. An example of this is given in the section “Getting Started: HPMDIXED Procedure” on page 3822.

---

**Default Output**

The following sections describe the output PROC HPMDIXED produces by default. This output is organized into various tables, and they are discussed in order of appearance.

**Model Information**

The “Model Information” table describes the model, some of the variables it involves, and the method used in fitting it. It also lists the method for computing the degrees of freedom.

The name of the “Model Information” table is ModelInfo.

**Class Level Information**

The “Class Level Information” table lists the first 20 levels of every variable specified in the CLASS statement. You should check this information to make sure the data are correct. You can adjust the order of the CLASS variable levels with the ORDER= option in the PROC HPMDIXED statement. The name of the “Class Level Information” table is ClassLevels.

**Dimensions**

The “Dimensions” table lists the sizes of relevant matrices. This table can be useful in determining CPU time and memory requirements. The name of the “Dimensions” table is “Dimensions.”

**Number of Observations**

The “Number of Observations” table shows the number of observations read from the data set and the number of observations used in fitting the model.

**Descriptive Statistics**

The “Descriptive Statistics” table lists simple statistics such as means and standard deviations for the dependent variable, for each covariate in the MODEL statement, and for the weight variable in the WEIGHT statement.
Iteration History

The “Iteration History” table describes the optimization of the residual log likelihood. The function to be minimized (the objective function) is $-2l$.

The name of the “Iteration History” table is IterHistory.

Covariance Parameter Estimates

The “Covariance Parameter Estimates” table contains the estimates of the parameters in $G$ and $R$. Their values are labeled in the “Cov Parm” table along with Subject and Group information if applicable. The estimates are displayed in the Estimate column.

The name of the “Covariance Parameter Estimates” table is CovParms.

Convergence Status

The “Convergence Status” table contains a status message that describes the reason the optimization terminated. The message is also written to the log. The name of the “Convergence Status” table is ConvergenceStatus. You can query the nonprinting numeric variable Status to check for a successful optimization. This is useful in batch processing, or when processing BY groups, such as in simulations. Successful optimizations are indicated by the value 0 for the Status variable.

Fit Statistics

The “Fit Statistics” table provides some statistics about the estimated mixed model.

In addition, the “Fit Statistics” table lists three information criteria: AIC, AICC, and BIC, all in smaller-is-better form. Expressions for these criteria are described under the IC= option on page 3828.

The name of the “Model Fitting Information” table is FitStatistics.

ODS Table Names

Each table created by PROC HPMIXED has a name associated with it, and you must use this name to reference the table when using ODS statements. These names are listed in Table 49.11.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Required Statement / Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>CholR</td>
<td>Cholesky root of blocks of the estimated $R$ matrix</td>
<td>REPEATED / RC</td>
</tr>
<tr>
<td>ClassLevels</td>
<td>Level information from the CLASS statement</td>
<td>Default output</td>
</tr>
<tr>
<td>Coef</td>
<td>$L$ matrix coefficients</td>
<td>E option in TEST, CONTRAST, ESTIMATE, or LSMEANS CONTRAST</td>
</tr>
<tr>
<td>Contrasts</td>
<td>Results from the CONTRAST statements</td>
<td>CONTRAST</td>
</tr>
<tr>
<td>ConvergenceStatus</td>
<td>Convergence status</td>
<td>Default</td>
</tr>
</tbody>
</table>
Table 49.11  (continued)

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Required Statement / Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>CovParms</td>
<td>Estimated covariance parameters</td>
<td>Default output</td>
</tr>
<tr>
<td>Diffs</td>
<td>Differences of LS-means</td>
<td>LSMEANS / DIFF (or PDIFF)</td>
</tr>
<tr>
<td>Dimensions</td>
<td>Dimensions of the model</td>
<td>Default output</td>
</tr>
<tr>
<td>Estimates</td>
<td>Results from ESTIMATE statements</td>
<td>ESTIMATE</td>
</tr>
<tr>
<td>FitStatistics</td>
<td>Fit statistics</td>
<td>Default</td>
</tr>
<tr>
<td>InvCholR</td>
<td>Inverse Cholesky root of blocks of the estimated R matrix</td>
<td>REPEATED / RCI=</td>
</tr>
<tr>
<td>InvR</td>
<td>Inverse of blocks of the estimated R matrix</td>
<td>REPEATED / RI=</td>
</tr>
<tr>
<td>IterHistory</td>
<td>Iteration history</td>
<td>Default output</td>
</tr>
<tr>
<td>LSMeans</td>
<td>LS-means</td>
<td>LSMEANS</td>
</tr>
<tr>
<td>MMEq</td>
<td>Mixed model equations</td>
<td>PROC HPMIXED MMEQ</td>
</tr>
<tr>
<td>ModelInfo</td>
<td>Model information</td>
<td>Default output</td>
</tr>
<tr>
<td>NObs</td>
<td>Number of observations read and used</td>
<td>Default output</td>
</tr>
<tr>
<td>OptInfo</td>
<td>Optimization information</td>
<td>Default output</td>
</tr>
<tr>
<td>OverallANOVA</td>
<td>ANOVA table for model without random effect</td>
<td>Default output for fixed models</td>
</tr>
<tr>
<td>ParameterEstimates</td>
<td>Fixed-effects solution</td>
<td>MODEL / SOLUTION</td>
</tr>
<tr>
<td>ParmSearch</td>
<td>Parameter search values</td>
<td>PARMS</td>
</tr>
<tr>
<td>Ranks</td>
<td>Ranks of matrices X, (XZ), and MMEQ</td>
<td>PROC HPMIXED RANKS</td>
</tr>
<tr>
<td>R</td>
<td>Blocks of the estimated R matrix</td>
<td>REPEATED / R=</td>
</tr>
<tr>
<td>RCorr</td>
<td>Correlation matrix from blocks of the estimated R matrix</td>
<td>REPEATED / RCORR=</td>
</tr>
<tr>
<td>SimpleStatistics</td>
<td>Descriptive statistics for dependent variable and covariate variables</td>
<td>PROC HPMIXED / SIMPLE</td>
</tr>
<tr>
<td>Slices</td>
<td>Tests of LS-means slices</td>
<td>LSMEANS / SLICE=</td>
</tr>
<tr>
<td>SolutionR</td>
<td>Random-effect solution vector</td>
<td>RANDOM / SOLUTION</td>
</tr>
<tr>
<td>Tests3</td>
<td>Type III tests of fixed effects</td>
<td>TEST</td>
</tr>
</tbody>
</table>

Examples: HPMIXED Procedure

Example 49.1: Ranking Many Random-Effect Coefficients

In analyzing models with random effects that have many levels, a frequent goal is to estimate and rank the predicted values of the coefficients corresponding to these levels. For example, in mixed models for animal breeding, the predicted coefficient of the random effect for each animal is referred to as the estimated breeding value (EBV) and animals with relatively high EBVs are chosen for breeding. This example demonstrates the use of the HPMIXED procedure for computing EBVs and their precision. Although other mixed modeling
tools in SAS/STAT can potentially compute EBVs, PROC HPMIXED is particularly suited for the large, sparse matrix calculations involved. The typical performance of the HPMIXED procedure and other tools for this problem is also discussed.

The data for this problem are generated by simulation. Suppose you are considering analyzing EBVs for animals on 15 farms, with about 100 animals of 5 different species on each farm. The following DATA step simulates data with this structure, where about 40 observations of the response variable Yield are made per animal:

```sas
%let NFarm = 15;
%let NAnimal = %eval(&NFarm*100);
data Sim;
  keep Species Farm Animal Yield;
  array BV{&NAnimal};
  array AnimalSpecies{&NAnimal};
  array AnimalFarm{&NAnimal};
  do i = 1 to &NAnimal;
    BV {i} = sqrt(4.0)*rannor(12345);
    AnimalSpecies{i} = 1 + int( 5 *ranuni(12345));
    AnimalFarm {i} = 1 + int(&NFarm*ranuni(12345));
  end;
  do i = 1 to 40*&NAnimal;
    Animal = 1 + int(&NAnimal*ranuni(12345));
    Species = AnimalSpecies{Animal};
    Farm = AnimalFarm {Animal};
    Yield = 1 + Species + Farm + BV{Animal} + sqrt(8.0)*rannor(12345);
    output;
  end;
run;
```

In this simulation, the true breeding value for each animal (BV1–BV1500) has a variance component of 4.0, while the level of background variance is 8.0.

In this type of experiment, the effect of Species and the interaction between Species and Farm are typically modeled as fixed effects, while the effect of Animal is modeled as a random effect. The following statements use the HPMIXED procedure to compute predictions for the Animal random effect and save them to the data set EBV. This data set is then sorted and the 10 animals with the highest EBVs are displayed.

```sas
ods listing close;
proc hpmixed data=Sim;
  class Species Farm Animal;
  model Yield = Species Farm*Species;
  random Animal/cl;
  ods output SolutionR=EBV;
run;
ods listing;

proc sort data=EBV;
  by descending estimate;
run;
proc print data=EBV(obs=10) noobs;
  var Animal Estimate StdErrPred Lower Upper;
run;
```
The preceding statements close the ODS listing destination for the duration of the PROC HPMIXED run. This avoids displaying the long random-effects solution table, since only the top few EBVs are of interest. Output 49.1.1 displays the EBVs of the top 10 animals, along with their precision and confidence bounds.

**Output 49.1.1** Estimated Breeding Values: Top 10 Animals

<table>
<thead>
<tr>
<th>Animal</th>
<th>Estimate</th>
<th>StdErrPred</th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>1294</td>
<td>5.9703</td>
<td>0.6317</td>
<td>4.7321</td>
<td>7.2085</td>
</tr>
<tr>
<td>1219</td>
<td>5.0081</td>
<td>0.6396</td>
<td>3.7544</td>
<td>6.2618</td>
</tr>
<tr>
<td>1054</td>
<td>4.9452</td>
<td>0.5874</td>
<td>3.7939</td>
<td>6.0966</td>
</tr>
<tr>
<td>758</td>
<td>4.9340</td>
<td>0.6196</td>
<td>3.7195</td>
<td>6.1485</td>
</tr>
<tr>
<td>986</td>
<td>4.9329</td>
<td>0.5767</td>
<td>3.8025</td>
<td>6.0633</td>
</tr>
<tr>
<td>1150</td>
<td>4.7444</td>
<td>0.5806</td>
<td>3.6064</td>
<td>5.8824</td>
</tr>
<tr>
<td>962</td>
<td>4.6651</td>
<td>0.5794</td>
<td>3.5294</td>
<td>5.8008</td>
</tr>
<tr>
<td>225</td>
<td>4.5294</td>
<td>0.6137</td>
<td>3.3266</td>
<td>5.7322</td>
</tr>
<tr>
<td>1252</td>
<td>4.5012</td>
<td>0.5686</td>
<td>3.3868</td>
<td>5.6157</td>
</tr>
<tr>
<td>1033</td>
<td>4.4971</td>
<td>0.6080</td>
<td>3.3054</td>
<td>5.6889</td>
</tr>
</tbody>
</table>

Notice that animal 1294 is ranked as the top animal based on its EBV, but the precision of this estimate, as measured by the standard error of prediction, is lower than that of other animals.

You can also use PROC MIXED and PROC GLIMMIX to compute EBVs, but the performance of these general mixed modeling procedures for this specialized kind of data and model is quite different from that of PROC HPMIXED. The MIXED and GLIMMIX procedures are engineered to have good performance properties across a broad class of models and analyses, a class much broader than what PROC HPMIXED can handle. The HPMIXED procedure, on the other hand, can have better performance, in terms of both memory and run time, for certain specialized models and analyses, of which the current example is one.

For this example, an equivalent PROC GLIMMIX approach can take twice as long to complete, and PROC MIXED three times as long. Precise relative timings are not feasible, since those of the MIXED and GLIMMIX procedures are sensitive to the speed of disk access for writing to and reading from the utility file that holds the underlying matrices. But the results on any system would be similar: for the limited class of models to which it applies, the sparse matrix representation that the HPMIXED procedure employs should provide better computational performance than a dense representation, in terms of both run time and memory use.

Moreover, for a given analysis, if the size of the problem is increased in such a way that the underlying matrices become sparser, the relative performance of PROC HPMIXED gets even better. As an illustration of this, Output 49.1.2 shows relative performance of the three procedures for simulated data as the number of farms increases. For this plot, each additional farm adds 500 levels of the Animal random effect to the model—a substantial number.
The vertical axis in Output 49.1.2 measures run time, but the units are omitted: relative performance is what counts, and that is expected to be fairly invariant to machine architecture. The output shows that while the performance of the MIXED and GLIMMIX procedures is relatively competitive with PROC HPMIXED for up to 3000 or 4000 animals, both procedures’ relative performance decreases as the number of animals increases into the tens of thousands.

As a caveat, note that PROC HPMIXED can be inefficient relative to PROC MIXED and PROC GLIMMIX for models and data that are not sparse, because it can take many times longer to invert a large, dense matrix by sparse techniques. For example, Output 49.1.3 shows relative performance of the three procedures for simulated data like the preceding, but where the fixed part of the model consists of an increasing number of continuous covariates and is thus dense.
Chapter 49: The HPMIXED Procedure

Output 49.1.3  Comparing Mixed Model Tools for Increasingly Dense Problems

As before, the HPMIXED procedure is more efficient than the MIXED and GLIMMIX procedures for few covariates, but when the fixed-effect calculations dominate the run time, PROC HPMIXED rapidly becomes relatively inefficient as the size of the dense fixed-effect matrix increases. Also note that while PROC MIXED is more efficient than PROC GLIMMIX for small to moderate numbers of covariates, PROC GLIMMIX has the best performance as the number of covariates get very large.

Example 49.2: Comparing Results from PROC HPMIXED and PROC MIXED

This example revisits the mixed model problem from the section “Getting Started: MIXED Procedure” on page 5218, in Chapter 65, “The MIXED Procedure,” with the data set shown in the following statements:

```r
data heights;
  input Family Gender$ Height @@;
  datalines;
  1 F 67   1 F 66   1 F 64   1 M 71   1 M 72   2 F 63   2 F 63  
  2 F 62   2 M 69   2 M 68   2 M 70   3 F 63   3 M 64   4 F 67  
  4 F 66   4 M 67   4 M 67   4 M 69   4 M 69
;```

Example 49.2: Comparing Results from PROC HPMIXED and PROC MIXED

The response variable Height measures the heights (in inches) of 18 individuals. The individuals are classified according to Family and Gender. The following statements fit a mixed model with random effects for Family and the Family*Gender interaction with the MIXED procedure:

```plaintext
proc mixed;
  class Family Gender;
  model Height = Gender / s;
  random Family Family*Gender / s;
run;
```

The “Iteration History” and “Fit Statistics” tables for the optimization in PROC MIXED are shown in Output 49.2.1. The MIXED procedure converges after six iterations and achieves a –2 restricted log likelihood of 71.02246.

**Output 49.2.1** Iteration History and Fit Statistics: MIXED Procedure

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Evaluations</th>
<th>-2 Res Log Like</th>
<th>Criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>74.11074833</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>71.51614003</td>
<td>0.01441208</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>71.13845990</td>
<td>0.00412226</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>71.03613556</td>
<td>0.00058188</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>71.02281757</td>
<td>0.00001689</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>71.02245904</td>
<td>0.00000002</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>71.02245869</td>
<td>0.00000000</td>
</tr>
</tbody>
</table>

**Output 49.2.2** Parameter Estimates and Solutions: MIXED Procedure

```
Output 49.2.2 displays the covariance parameter estimates and the solutions for the fixed and random effects. Because the fixed-effect model contains a classification effect (Gender) and an intercept, the \(X'X\) matrix is singular. Only two fixed-effect parameters can be estimated in this model. The MIXED procedure, relying on a sweep operation in the order in which effects enter the model, determines that the last column of the \(X'X\) matrix is a linear function of previous columns. Consequently, the coefficient for the second level of the Gender variable is zero.

**Output 49.2.2** Parameter Estimates and Solutions: MIXED Procedure

<table>
<thead>
<tr>
<th>Covariance Parameter Estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cov Parm</td>
</tr>
<tr>
<td>Family</td>
</tr>
<tr>
<td>Family*Gender</td>
</tr>
<tr>
<td>Residual</td>
</tr>
</tbody>
</table>
The “Type 3 Tests of Fixed Effects” table in Output 49.2.3 is produced by the MIXED procedure by default.

**Output 49.2.3** Test of Gender Effect

<table>
<thead>
<tr>
<th>Effect</th>
<th>Gender</th>
<th>Num</th>
<th>Den</th>
<th>DF</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td></td>
<td>1</td>
<td>3</td>
<td>7.95</td>
<td>0.0667</td>
<td></td>
</tr>
</tbody>
</table>

The same linear mixed model is fit with the HPMIXED procedure with the following statements:

```plaintext
proc hpmixed;
  class Family Gender;
  model Height = Gender / s;
  random Family Family*Gender / s;
  test gender;
run;
```

Output 49.2.4 displays the “Iteration History” and “Fit Statistics” tables. The HPMIXED procedure, with its default quasi-Newton algorithm, achieves the same –2 restricted log likelihood as the MIXED procedure (71.02246; see Output 49.2.1).
**Output 49.2.4** Iteration History and Fit Statistics: HPMIXED Procedure

**The HPMIXED Procedure**

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Evaluations</th>
<th>Objective Function</th>
<th>Change</th>
<th>Max Gradient</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>71.023177956</td>
<td>.</td>
<td>0.034074</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>71.022519936</td>
<td>0.00065802</td>
<td>0.007839</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>71.022477283</td>
<td>0.00004265</td>
<td>0.004674</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>71.0224587</td>
<td>0.00001858</td>
<td>0.000168</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>71.022458689</td>
<td>0.00000001</td>
<td>3.28E-6</td>
</tr>
</tbody>
</table>

**Fit Statistics**

- -2 Res Log Likelihood: 71.02246
- AIC (Smaller is Better): 77.02246
- AICC (Smaller is Better): 79.02246
- BIC (Smaller is Better): 75.18134
- CAIC (Smaller is Better): 78.18134
- HQIC (Smaller is Better): 72.98226

**Output 49.2.5** displays the results that correspond to those in **Output 49.2.2** in the MIXED procedure.

**Output 49.2.5** Parameter Estimates and Solutions: HPMIXED Procedure

<table>
<thead>
<tr>
<th>Cov Parm</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Family</td>
<td>2.4010</td>
</tr>
<tr>
<td>Family*Gender</td>
<td>1.7657</td>
</tr>
<tr>
<td>Residual</td>
<td>2.1668</td>
</tr>
</tbody>
</table>

**Solution for Fixed Effects**

<table>
<thead>
<tr>
<th>Effect</th>
<th>Gender</th>
<th>Estimate</th>
<th>Standard Error</th>
<th>DF</th>
<th>t Value</th>
<th>Pr &gt;</th>
<th>t</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td></td>
<td>0</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td>F</td>
<td>64.8493</td>
<td>1.1477</td>
<td>16</td>
<td>56.50</td>
<td>&lt;.001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td>M</td>
<td>68.2114</td>
<td>1.1477</td>
<td>16</td>
<td>59.43</td>
<td>&lt;.001</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A number of points are noteworthy in comparing the results from the procedures. The covariance parameter estimates are the same, yet the solutions for the fixed effects differ. In fact, both solutions are correct. Solving a sparse system of linear equations requires reordering of the mixed model equations to minimize memory consumption in the factorization process. As a consequence, the order in which singularities are detected can differ from the order in which effects enter the model. Mathematically, the two sets of solutions simply correspond to different choices for the generalized inverse in solving a singular linear system. See the sections “Generalized Inverse Matrices” on page 42 and “Linear Model Theory” on page 51, in Chapter 3, “Introduction to Statistical Modeling with SAS/STAT Software,” for more information about the role and importance of generalized inverses in linear model analysis.

Although the two sets of solutions for the fixed effects correspond to different choices of generalized inverses, many important results are invariant to the choice of the $g$-inverse. For example, the solutions for the random effects in Output 49.2.5 and Output 49.2.2 are identical. Also, the test for the Gender effect yields the same $F$ value in both analyses (compare Output 49.2.6 and Output 49.2.3). However, note that the $p$-values associated with both $F$ tests and $t$ tests differ between the two procedures. This is due to their different default methods for computing the degrees of freedom. For this model, the HPMIXED procedure uses the residual method to determine the denominator degrees of freedom for tests of fixed effects, whereas the MIXED procedure uses the containment method. The containment method is order-dependent, and thus not available in the HPMIXED procedure.

**Output 49.2.6** Parameter Estimates and Solutions: HPMIXED Procedure

| Effect   | Gender | Family | Estimate | Std Err | Pred DF | t Value | Pr > |t| |
|----------|--------|--------|----------|---------|---------|---------|------|---|
| Family   | 1      | 1.2680 | 1.1201   | 16      | 1.13    | 0.2743  |      |  
| Family   | 2      | 0.08980| 1.1121   | 16      | 0.08    | 0.9366  |      |  
| Family   | 3      | -1.6660| 1.1712   | 16      | -1.42   | 0.1741  |      |  
| Family   | 4      | 0.3082 | 1.1201   | 16      | 0.28    | 0.7867  |      |  
| Family*Gender | F | 1 | -0.3198 | 1.0810 | 16 | -0.30 | 0.7712 |      |  
| Family*Gender | M | 1 | 1.2523  | 1.0933 | 16 | 1.15  | 0.2689 |      |  
| Family*Gender | F | 2 | -0.4299 | 1.0774 | 16 | -0.40 | 0.6951 |      |  
| Family*Gender | M | 2 | 0.4959  | 1.0774 | 16 | 0.46  | 0.6515 |      |  
| Family*Gender | F | 3 | -0.08229| 1.1409 | 16 | -0.07 | 0.9434 |      |  
| Family*Gender | M | 3 | -1.1429 | 1.1409 | 16 | -1.00 | 0.3314 |      |  
| Family*Gender | F | 4 | 0.8320  | 1.0933 | 16 | 0.76  | 0.4577 |      |  
| Family*Gender | M | 4 | -0.6053 | 1.0810 | 16 | -0.56 | 0.5832 |      |  

**Output 49.2.6** Type III Tests of Fixed Effects: HPMIXED Procedure

<table>
<thead>
<tr>
<th>Effect</th>
<th>Num DF</th>
<th>Den DF</th>
<th>F Value</th>
<th>Pr &gt;</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td>1</td>
<td>16</td>
<td>7.95</td>
<td>0.0123</td>
<td></td>
</tr>
</tbody>
</table>

This output shows the Type III tests of fixed effects for the Gender effect, with a $F$ value of 7.95 and a $p$-value of 0.0123.
Example 49.3: Using PROC GLIMMIX for Further Analysis of PROC HPMIXED Fit

The HPMIXED procedure handles only a subset of the analyses of the GLIMMIX procedure. However, you can use the HPMIXED procedure to accelerate your GLIMMIX procedure analyses for large problems. The idea is to use PROC HPMIXED to maximize the likelihood and produce parameter estimates more quickly than PROC GLIMMIX, and then to pass these parameter estimates to PROC GLIMMIX for some further analysis that is not available within PROC HPMIXED.

This example revisits the mixed model problem from the section “Getting Started: HPMIXED Procedure” on page 3822 to illustrate how to obtain the covariance estimates from the HPMIXED procedure and, in turn, how to use these estimates in PROC GLIMMIX’s PARMS statement. The following statements again simulate data from animals of different species on different farms:

```sas
data Sim;
  keep Species Farm Animal Yield;
  array AnimalEffect{3000};
  array AnimalSpecies{3000};
  array AnimalFarm{3000};
  do i = 1 to 3000;
    AnimalEffect{i} = sqrt(4.0)*rannor(12345);
    AnimalSpecies{i} = 1 + int(5*ranuni(12345));
    AnimalFarm{i} = 1 + int(10*ranuni(12345));
  end;
  do i = 1 to 40000;
    Animal = 1 + int(3000*ranuni(12345));
    Species = AnimalSpecies{Animal};
    Farm = AnimalFarm{Animal};
    Yield = 1 + Species + int(Farm/2) + AnimalEffect{Animal} + sqrt(8.0)*rannor(12345);
    output;
  end;
run;
```

Note that in the preceding DATA step program, certain pairs of farms are simulated to have the same effect on yield. Suppose that your goal is to determine which farms are significantly different. While the HPMIXED procedure has an LSMEANS statement, it has no options for multiple comparisons. The following statements first use the HPMIXED procedure to obtain the covariance estimates, saving them in the SAS data set HPMEstimate. Then the GLIMMIX procedure is executed with the PARMS statement to initialize the parameter values from the data set HPMEstimate and with the HOLD= and NOITER options to prevent further optimization iterations. The LSMEANS statement is used in PROC GLIMMIX to perform multiple comparisons of the LS-means for farms, and the results are displayed as a so-called diffogram.

```sas
proc hpmixed data=Sim;
  class Species Farm Animal;
  model Yield = Farm|Species;
  random Animal;
  test Species Species*Farm;
  ods output CovParms=HPMEstimate;
run;
```
ods graphics on;
proc glimmix data=Sim;
   class Species Farm Animal;
   model Yield = Farm|Species;
   random int/sub=Animal;
   parms /pdata=HPMEstimate hold=1,2 noiter;
   lsmeans Farm / pdiff=all plot=diffplot;
run;

The iteration histories for the two procedures are shown in Output 49.3.1 and Output 49.3.2. Whereas PROC HPMIXED requires several iterations in order to converge, PROC GLIMMIX “converges” to the same value in one step, with no iteration since the options HOLD= and NOITER are used.

Output 49.3.1  Iteration History for the HPMIXED Procedure

The HPMIXED Procedure

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Evaluations</th>
<th>Objective Function</th>
<th>Change</th>
<th>Max Gradient</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>202516.66891</td>
<td>.0841954</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>202516.66887</td>
<td>0.00004385</td>
<td>0.000641</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>202516.66887</td>
<td>-0.00000000</td>
<td>0.000641</td>
</tr>
</tbody>
</table>

Output 49.3.2  Iteration History for the GLIMMIX Procedure

The GLIMMIX Procedure

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Restarts</th>
<th>Evaluations</th>
<th>Objective Function</th>
<th>Change</th>
<th>Max Gradient</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>4</td>
<td>202516.66887</td>
<td>.</td>
<td>0</td>
</tr>
</tbody>
</table>

The graphical multiple-comparisons analysis for the LS-means of farms is shown in Output 49.3.3. It confirms the pairwise equalities between farm effects with which the data were simulated.
Output 49.3.3  LS-Means Plot of Pairwise Farm Differences

Example 49.4: Mixed Model Analysis of Microarray Data

Microarray experiments are an advanced genomic technique used in the discovery of new treatments for diseases. Microarray analysis allows for the detection of tens of thousands of genes in a single DNA sample. A microarray is a glass slide or membrane that has been spotted or “arrayed” with DNA fragments or oligonucleotides representing specific genes. The response of the gene detected by a spot is proportional to the intensity of fluorescence associated with that spot. These gene responses can indicate associations with disease conditions, but they can also be affected by systematic biases and different treatments such as sex and genotypes. Statistical models for microarray data attempt to assess the significance and magnitude of gene effects across treatments while adjusting for these systematic biases and to evaluate the significance of differences between treatments.

There are two statistical approaches frequently used in mixed model analysis for microarray data. The first approach is to fit multiple gene-specific models to data normalized for systematic biases (Wolfinger et al. 2001; Gibson and Wolfinger 2004). This approach is based on assuming that
the biases are independent from the gene effects. If this assumption is untenable, then a second approach fits a single model that combines both the systematic biases and the gene effects (Kerr, Martin, and Churchill 2000; Churchill 2002; Littell et al. 2006). When the number of genes is very large, several hundreds to tens of thousands, this is an analysis for which the sparse matrix approach implemented in the HPMIXED procedure is well suited.

The following SAS statements simulate a microarray experiment with a so-called loop design structure, which is commonly used in such studies. There are 500 genes, each gene occurs in 6 arrays, and each array has 2 dyes.

```sas
%let narray = 6;
%let ndye = 2;
%let nrow = 4;
%let ngene = 500;
%let ntrt = 6;
%let npin = 4;
%let ndip = 4;
%let no = %eval(&ndye*&nrow*&ngene);
%let tno = %eval(&narray*&no);

data microarray;
  keep Gene MArray Dye Trt Pin Dip log2i;
  array PinDist{&tno};
  array DipDist{&tno};
  array GeneDist{&tno};
  array ArrayEffect{&narray};
  array ArrayGeneEffect{%eval(&narray*&ngene)};
  array ArrayDipEffect{%eval(&narray*&ndip)};
  array ArrayPinEffect{%eval(&narray*&npin)};
  do i = 1 to &tno;
    PinDist{i} = 1 + int(&npin*ranuni(12345));
    DipDist{i} = 1 + int(&ndip*ranuni(12345));
    GeneDist{i} = 1 + int(&ngene*ranuni(12345));
  end;
  igene = 0;
  idip = 0;
  ipin = 0;
  do i = 1 to &narray;
    ArrayEffect{i} = sqrt(0.014)*rannor(12345);
    do j = 1 to &ngene;
      igene = igene+1;
      ArrayGeneEffect{igene} = sqrt(0.0017)*rannor(12345);
    end;
    do j = 1 to &ndip;
      idip = idip + 1;
      ArrayDipEffect{idip} = sqrt(0.0033)*rannor(12345);
    end;
    do j = 1 to &npin;
      ipin = ipin + 1;
      ArrayPinEffect{ipin} = sqrt(0.037)*rannor(12345);
  end;
```

Example 49.4: Mixed Model Analysis of Microarray Data

i = 0;
do MArray = 1 to &narray;
   do Dye = 1 to &ndye;
      do Row = 1 to &nrow;
         do k = 1 to &ngene;
            if MArray=1 and Dye = 1 then do;
               Trt = 0;
               trtc = 0;
            end;
            else do;
               if trtc >= &no then trtc = 0;
               if trtc = 0 then do;
                  Trt = Trt + 1;
                  if Trt >= &ntrt then do;
                     Trt = 0;
                     trtc = 0;
                  end;
               end;
               trtc = trtc + 1;
         end;
         i = i + 1;
         Pin = PinDist{i};
         Dip = DipDist{i};
         Gene = GeneDist{i};
         a = ArrayEffect{MArray};
         ag = ArrayGeneEffect{(MArray-1) *&ngene+Gene};
         ad = ArrayDipEffect{(MArray-1) *&ndip+Dip};
         ap = ArrayPinEffect{(MArray-1) *&npin+Pin};
         log2i = 1 +
         + Dye
         + Trt
         + Gene/1000.0
         + Dye*Gene/1000.0
         + Trt*Gene/1000.0
         + Pin
         + a
         + ag
         + ad
         + ap
         + sqrt(0.02)*rannor(12345);
         output;
      end;
   end;
end;
run;
A linear mixed model for fitting the log intensity data $Y_{ijkmn}$ from such a design is described by Littell et al. (2006) as follows:

$$Y_{ijkmn} =$$  

<table>
<thead>
<tr>
<th>Fixed Effects</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$</td>
<td>Overall mean</td>
</tr>
<tr>
<td>$\lambda_i$</td>
<td>Gene</td>
</tr>
<tr>
<td>$\tau_j$</td>
<td>Treatment</td>
</tr>
<tr>
<td>$\delta_k$</td>
<td>Dye</td>
</tr>
<tr>
<td>$(\tau \lambda)_{ij}$</td>
<td>Treatment-by-gene</td>
</tr>
<tr>
<td>$(\delta \lambda)_{jk}$</td>
<td>Dye-by-gene</td>
</tr>
<tr>
<td>$p_r$</td>
<td>Pin</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Random Effects</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_m$</td>
<td>Microarray</td>
</tr>
<tr>
<td>$(a \lambda)_{im}$</td>
<td>Microarray-by-gene</td>
</tr>
<tr>
<td>$d(a)_{mn}$</td>
<td>Dip-within-microarray</td>
</tr>
<tr>
<td>$(a \rho)_{mr}$</td>
<td>Microarray-by-pin</td>
</tr>
<tr>
<td>$e_{ijkmn}$</td>
<td>Residual noise</td>
</tr>
</tbody>
</table>

You can use the HPMIXED procedure with the following statements to fit this model:

```plaintext
proc hpmixed data=microarray;  
class marray dye trt gene pin dip;  
model log2i = dye trt gene dye*gene trt*gene pin;  
random marray marray*gene dip(marray) pin*marray;  
test trt;  
run;
```

The “Dimensions” table shown in Output 49.4.1 indicates that this is a very large model, with 4512 columns in $X$ matrix and 3054 columns in $Z$ matrix. It will be computationally very inefficient to fit this model by using dense matrix methods; the sparse matrix approach of the HPMIXED procedure is of critical importance.

**Output 49.4.1** Mixed Model Dimensions

<table>
<thead>
<tr>
<th>The HPMIXED Procedure</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions</td>
<td></td>
</tr>
<tr>
<td>G-side Cov. Parameters</td>
<td>4</td>
</tr>
<tr>
<td>R-side Cov. Parameters</td>
<td>1</td>
</tr>
<tr>
<td>Columns in X</td>
<td>4513</td>
</tr>
<tr>
<td>Columns in Z</td>
<td>3054</td>
</tr>
<tr>
<td>Subjects (Blocks in V)</td>
<td>1</td>
</tr>
</tbody>
</table>

The $p$-value in Output 49.4.2 indicates that there are significant differences between treatments.

**Output 49.4.2** Type III Tests of Fixed Effects

<table>
<thead>
<tr>
<th>Type III Tests of Fixed Effects</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Effect</td>
<td>Num DF</td>
</tr>
<tr>
<td>Trt</td>
<td>5</td>
</tr>
</tbody>
</table>
Example 49.5: Repeated Measures

The following data are from Pothoff and Roy (1964) and consist of growth measurements for 11 girls and 16 boys at ages 8, 10, 12, and 14. Some of the observations are suspect (for example, the third observation for person 20); however, all of the data are used here for comparison purposes.

The analysis strategy employs a linear growth curve model for the boys and girls in addition to a variance-covariance model that incorporates correlations for all of the observations that arise from the same person. The PROC HP MIXED statements to fit an unstructured variance matrix are as follows:

```
data pr;
  input Person Gender $ y1 y2 y3 y4;
  y=y1; Time=1; Age=8; output;
  y=y2; Time=2; Age=10; output;
  y=y3; Time=3; Age=12; output;
  y=y4; Time=4; Age=14; output;
  drop y1-y4;
  datalines;
1   F 21.0 20.0 21.5 23.0
2   F 21.0 21.5 24.0 25.5
3   F 20.5 24.0 24.5 26.0
4   F 23.5 24.5 25.0 26.5
5   F 21.5 23.0 22.5 23.5
6   F 20.0 21.0 21.0 22.5
7   F 21.5 22.5 23.0 25.0
8   F 23.0 23.0 23.5 24.0
9   F 20.0 21.0 22.0 21.5
10  F 16.5 19.0 19.0 19.5
11  F 24.5 25.0 28.0 28.0
12  M 26.0 25.0 29.0 31.0
13  M 21.5 22.5 23.0 26.5
14  M 23.0 22.5 24.0 27.5
15  M 25.5 27.5 26.5 27.0
16  M 20.0 23.5 22.5 26.0
17  M 24.5 25.5 27.0 28.5
18  M 22.0 22.0 24.5 26.5
19  M 24.0 21.5 24.5 25.5
20  M 23.0 20.5 31.0 26.0
21  M 27.5 28.0 31.0 31.5
22  M 23.0 23.0 23.5 25.0
23  M 21.5 23.5 24.0 28.0
24  M 17.0 24.5 26.0 29.5
25  M 22.5 25.5 25.5 26.0
26  M 23.0 24.5 26.0 30.0
27  M 22.0 21.5 23.5 25.0
;
proc hpmixed data=pr;
  class Person Gender Time;
  model y = Gender Age Gender*Age;
  test Gender Age Gender*Age;
  repeated Time / type=un subject=Person r;
run;
```
The **MODEL** statement first lists the dependent variable $Y$. The fixed effects are then listed after the equal sign. The variable **Gender** requests a different intercept for the girls and boys, **Age** models an overall linear growth trend, and **Gender*Age** makes the slopes different over time. It is actually not necessary to specify **Age** separately, but doing so enables PROC HPMIXED to carry out a test for heterogeneous slopes.

The **REPEATED** statement contains a *repeated-effect* **Time**. The **TYPE=UN** option models the covariance as an unstructured block for each **SUBJECT=Person**. Each of the 27 subjects has a maximum of four observations. Therefore, the $R$ matrix is block diagonal with 27 blocks, each block consisting of identical $4 \times 4$ unstructured matrices. The 10 parameters of these unstructured blocks make up the covariance parameters estimated by restricted maximum likelihood. The **R=** option requests that the first block of $R$ be displayed.

The results from this analysis are shown in **Output 49.5.1** through **Output 49.5.5**.

**Output 49.5.1**  Repeated Measures Analysis

<table>
<thead>
<tr>
<th>The HPMIXED Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions</td>
</tr>
<tr>
<td>G-side Cov. Parameters</td>
</tr>
<tr>
<td>R-side Cov. Parameters</td>
</tr>
<tr>
<td>Columns in X</td>
</tr>
<tr>
<td>Columns in Z per Subject</td>
</tr>
<tr>
<td>Subjects (Blocks in V)</td>
</tr>
</tbody>
</table>

In **Output 49.5.1**, the 10 covariance parameters result from the $4 \times 4$ unstructured blocks of $R$. There is no $Z$ matrix for this model.

**Output 49.5.2**  Repeated Measures Analysis (continued)

<table>
<thead>
<tr>
<th>Number of Observations Read</th>
<th>108</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Observations Used</td>
<td>108</td>
</tr>
</tbody>
</table>
The 17 quasi-Newton iterations are used to find the maximum likelihood estimates (Output 49.5.2).

Output 49.5.3 Repeated Measures Analysis (continued)

The 4×4 matrix in Output 49.5.3 is the estimated unstructured covariance matrix. It is the estimate of the first block of \( R \), and the other 26 blocks all have the same estimate.
The “Covariance Parameter Estimates” table in Output 49.5.4 lists the 10 estimated covariance parameters in order; note their correspondence to the first block of R displayed in Output 49.5.3. The parameter estimates are labeled according to their location in the block in the Cov Parm column, and all of these estimates are associated with Person as the subject effect.

### Output 49.5.5 Repeated Measures Analysis (continued)

<table>
<thead>
<tr>
<th>Covariance Parameter Estimates</th>
<th>Cov Parm</th>
<th>Subject</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>UN(1,1)</td>
<td>Person</td>
<td>5.4252</td>
<td></td>
</tr>
<tr>
<td>UN(2,1)</td>
<td>Person</td>
<td>2.7092</td>
<td></td>
</tr>
<tr>
<td>UN(2,2)</td>
<td>Person</td>
<td>4.1906</td>
<td></td>
</tr>
<tr>
<td>UN(3,1)</td>
<td>Person</td>
<td>3.8411</td>
<td></td>
</tr>
<tr>
<td>UN(3,2)</td>
<td>Person</td>
<td>2.9745</td>
<td></td>
</tr>
<tr>
<td>UN(3,3)</td>
<td>Person</td>
<td>6.2632</td>
<td></td>
</tr>
<tr>
<td>UN(4,1)</td>
<td>Person</td>
<td>2.7151</td>
<td></td>
</tr>
<tr>
<td>UN(4,2)</td>
<td>Person</td>
<td>3.1317</td>
<td></td>
</tr>
<tr>
<td>UN(4,3)</td>
<td>Person</td>
<td>4.1332</td>
<td></td>
</tr>
<tr>
<td>UN(4,4)</td>
<td>Person</td>
<td>4.9862</td>
<td></td>
</tr>
</tbody>
</table>

The “Type III Tests of Fixed Effects” table in Output 49.5.5 displays Type III tests for all of the fixed effects. These tests are partial in the sense that they account for all of the other fixed effects in the model.

Since the different levels of the repeated effect represent different years, it is natural to try fitting a time series model to the data within each subject. To obtain time series structures in R, you can replace TYPE=UN with TYPE=AR(1) to obtain the first-order autoregressive covariance matrices. For example, the statements to fit an AR(1) structure are as follows:

```r
proc hpmixed data=pr;
  class Person Gender Time;
  model y = Gender Age Gender*Age;
  repeated Time / type=ar(1) sub=Person r;
run;
```

The estimated AR(1) structure covariance matrix of the first block of R is shown in Output 49.5.6.
**Output 49.5.6** Repeated Measures Analysis

**The HPMIXED Procedure**

<table>
<thead>
<tr>
<th>Row</th>
<th>Col1</th>
<th>Col2</th>
<th>Col3</th>
<th>Col4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.2144</td>
<td>3.2563</td>
<td>2.0335</td>
<td>1.2699</td>
</tr>
<tr>
<td>2</td>
<td>3.2563</td>
<td>5.2144</td>
<td>3.2563</td>
<td>2.0335</td>
</tr>
<tr>
<td>3</td>
<td>2.0335</td>
<td>3.2563</td>
<td>5.2144</td>
<td>3.2563</td>
</tr>
<tr>
<td>4</td>
<td>1.2699</td>
<td>2.0335</td>
<td>3.2563</td>
<td>5.2144</td>
</tr>
</tbody>
</table>

**References**


Subject Index

Akaike's information criterion
  HPMIXED procedure, 3828, 3863
Akaike's information criterion (finite sample corrected version)
  HPMIXED procedure, 3828, 3863
alpha level
  HPMIXED procedure, 3838, 3840, 3842, 3849
AR(1) structure
  HPMIXED procedure, 3852
autoregressive structure
  example (HPMIXED), 3882
  HPMIXED procedure, 3852
boundary constraints
  HPMIXED procedure, 3847, 3849
chi-square test
  HPMIXED procedure, 3834
Cholesky
covariance structure (HPMIXED), 3852
  root (HPMIXED), 3852
class level
  HPMIXED procedure, 3829
compound symmetry structure
  HPMIXED procedure, 3853
confidence limits
  HPMIXED procedure, 3838, 3840, 3842, 3845
least squares means (HPMIXED), 3840
solution for random effects (HPMIXED), 3850
constraints
  boundary (HPMIXED), 3847, 3849
contrast specification
  HPMIXED procedure, 3833
contrasts
  HPMIXED procedure, 3833
convergence status
  HPMIXED procedure, 3863
correlation
  estimates (HPMIXED), 3855
correlations of least squares means
  HPMIXED procedure, 3841
covariance parameter estimates
  HPMIXED procedure, 3863
covariance structure
  autoregressive (HPMIXED), 3852
  Cholesky type (HPMIXED), 3852
  compound symmetry (HPMIXED), 3853
  equi-correlation (HPMIXED), 3853
  examples (HPMIXED), 3851
  heterogeneous compound symmetry (HPMIXED), 3853
  heterogeneous uniform correlation (HPMIXED), 3854
  positive (semi-)definite, 3852
  uniform correlation (HPMIXED), 3853
  unstructured (HPMIXED), 3854
  variance components (HPMIXED), 3854
covariance structures
  examples (HPMIXED), 3879
covariances of least squares means
  HPMIXED procedure, 3841
degrees of freedom
  HPMIXED procedure, 3834, 3838, 3841, 3843
  infinite (HPMIXED), 3840, 3843
  residual method (HPMIXED), 3843
descriptive statistics
  mixed model (HPMIXED), 3862
dimensions
  HPMIXED procedure, 3830
EM-REML
  HPMIXED procedure, 3859
estimability
  HPMIXED procedure, 3833, 3835, 3839, 3843
estimates
  HPMIXED procedure, 3837
estimation methods
  HPMIXED procedure, 3829
examples, HPMIXED
  animal breeding data, 3822
  autoregressive structure, R-side, 3882
  getting started, 3822
  least squares means, differences against control, 3841
  least squares means, slice, 3842
  many fixed and random effects, 3822
  NOITER option for covariance parameters, 3847
  Pothoff and Roy growth measurements, 3879
  slice F test, 3842
  starting values and BY groups, 3848
  starting values from data set, 3848
  subject-specific R matrices, 3855
fixed effects
  HPMIXED procedure, 3842
G matrix
HPMIXED procedure, 3849, 3857
GLIMMIX procedure
  least squares means, 3839
grid search
  HPMIXED procedure, 3846
Hannan-Quinn information criterion
  HPMIXED procedure, 3828
heterogeneity
  HPMIXED procedure, 3850, 3855
heterogeneous compound symmetry
  HPMIXED procedure, 3853
heterogeneous uniform correlation structure
  HPMIXED procedure, 3854
HPMIXED procedure
  Akaike's information criterion, 3828, 3863
  Akaike's information criterion (finite sample
    corrected version), 3828, 3863
  alpha level, 3838, 3840, 3842, 3849
  AR(1) structure, 3852
  autoregressive structure, 3852, 3882
  average information, 3820, 3859
  basic features, 3818
  BLUE, 3827
  BLUP, 3827
  BLUPs, 3850
  boundary constraints, 3847, 3849
  chi-square test, 3834, 3843
  Cholesky covariance structure, 3852
  Cholesky root, 3852
  class level, 3829
  comparing HPMIXED and MIXED, 3868
  compound symmetry structure, 3853
  confidence interval, 3849
  confidence limits, 3838, 3840, 3842, 3845, 3850
  conjugate gradient algorithm, 3820
  continuous effects, 3850, 3855, 3856
  contrast specification, 3833
  contrasts, 3833
  convergence status, 3863
  correlation estimates, 3855
  correlations of least squares means, 3841
  covariance parameter estimates, 3863
  covariance structure, 3851, 3879
  covariances of least squares means, 3841
  degrees of freedom, 3834, 3837, 3838, 3840, 3841, 3843
  dimensions, 3830
  effect name length, 3829
  EM-REML, 3859
  estimability, 3833, 3835, 3839, 3840, 3843
  estimates, 3837
  estimation methods, 3829
  expected information, 3859
  first and second derivatives, 3859
  fitting information, 3863
  fixed effects, 3842
  fixed-effects parameters, 3843
  G matrix, 3849, 3857
  grid search, 3846
  Hannan-Quinn information criterion, 3828
  heterogeneity, 3850, 3855
  heterogeneous compound symmetry, 3853
  heterogeneous uniform correlation structure, 3854
  hypothesis tests, 3861
  infinite degrees of freedom, 3838, 3840, 3843
  information criteria, 3828
  initial values, 3846
  input data sets, 3828
  intercept effect, 3843, 3849
  introductory example, 3822
  iteration details, 3829
  iterations, 3863
  L matrices, 3833, 3840
  least squares means, 3841
  likelihood computation, 3858
  microarray data, 3875
  mixed model, 3842
  mixed model equations, 3829
  model assumptions, 3857
  model information, 3830
  multiple comparisons of least squares means, 3841
  number of observations, 3830
  ODS table names, 3863
  parameter constraints, 3847
  positive definiteness, 3852
  profiling residual variance, 3830
  R matrix, 3855, 3857
  random effects, 3849
  random-effects parameter, 3850
  repeated effects, 3854
  repeated measures, 3879
  residual likelihood, 3829
  residual method, 3843
  residual variance tolerance, 3831
  restricted maximum likelihood, 3829
  rounding error, 3861
  Schwarz's Bayesian information criterion, 3828, 3863
  simple effects, 3842
  singularity, 3831
  sparse matrix storage, 3860
  sparse matrix techniques, 3820, 3860
  starting values, 3859
  subject effect, 3850, 3856
  summary of commands, 3825
  table names, 3863
Syntax Index

ALLSTATS option
   OUTPUT statement (HPMIXED), 3845

ALPHA= option
   ESTIMATE statement (HPMIXED), 3838
   LSMEANS statement (HPMIXED), 3840
   MODEL statement (HPMIXED), 3842
   OUTPUT statement (HPMIXED), 3845
   RANDOM statement (HPMIXED), 3849

BLUP= option
   PROC HPMIXED statement, 3827

BY statement
   HPMIXED procedure, 3831

CHISQ option
   CONTRAST statement (HPMIXED), 3834
   TEST statement (HPMIXED), 3857

CL option
   ESTIMATE statement (HPMIXED), 3838
   LSMEANS statement (HPMIXED), 3840
   MODEL statement (HPMIXED), 3842
   OUTPUT statement (HPMIXED), 3845
   RANDOM statement (HPMIXED), 3850

CLASS statement
   HPMIXED procedure, 3862
   HPMIXED procedure, 3832

CONTRAST statement
   HPMIXED procedure, 3833

COV option
   LSMEANS statement (HPMIXED), 3841

DATA= option
   PROC HPMIXED statement, 3828

DDF= option
   MODEL statement (HPMIXED), 3843

DDFM= option
   MODEL statement (HPMIXED), 3843

DF= option
   CONTRAST statement (HPMIXED), 3834
   ESTIMATE statement (HPMIXED), 3838
   LSMEANS statement (HPMIXED), 3841

DIFF option
   LSMEANS statement (HPMIXED), 3841

DIVISOR= option
   ESTIMATE statement (HPMIXED), 3838

E option
   CONTRAST statement (HPMIXED), 3835
   ESTIMATE statement (HPMIXED), 3839
   LSMEANS statement (HPMIXED), 3842
   TEST statement (HPMIXED), 3856

E3 option
   TEST statement (HPMIXED), 3857

EFFECT statement
   HPMIXED procedure, 3835

ESTIMATE statement
   HPMIXED procedure, 3837

GLIMMIX procedure
   LSMEANS statement
   PDIF option, 3841

GROUP option
   CONTRAST statement (HPMIXED), 3835
   ESTIMATE statement (HPMIXED), 3839

GROUP= option
   RANDOM statement (HPMIXED), 3850
   REPEATED statement (HPMIXED), 3855

HOLD= option
   PARMS statement (HPMIXED), 3847

HPMIXED procedure
   CONTRAST statement, 3833
   ESTIMATE statement, 3837
   ID statement, 3839
   LSMEANS statement, 3839
   MODEL statement, 3842
   NLOPTIONS statement, 3843
   OUTPUT statement, 3844
   PARMS statement, 3846
   PROC HPMIXED statement, 3826
   RANDOM statement, 3849
   REPEATED statement, 3854
   TEST statement, 3856
   WEIGHT statement, 3857

HPMIXED procedure, BY statement, 3831

HPMIXED procedure, CLASS statement, 3862

HPMIXED procedure, CONTRAST statement, 3833
   CHISQ option, 3834
   DF= option, 3834
   E option, 3835
   GROUP option, 3835
   SINGULAR= option, 3835
   SUBJECT= option, 3835

HPMIXED procedure, ESTIMATE statement, 3837
   ALPHA= option, 3838
   CL option, 3838
   DF= option, 3838
   DIVISOR= option, 3838
ID statement
   HPMIXED procedure, 3839
INFOCRIT= option
   PROC HPMIXED statement, 3828
ITDETAILS option
   PROC HPMIXED statement, 3829
LCL= option
   OUTPUT statement (HPMIXED), 3844
LOGNOTE option
   PROC HPMIXED statement, 3829
LOWERB= option
   PARMS statement (HPMIXED), 3847
LSMEANS statement
   HPMIXED procedure, 3839
MAXCLPRINT= option
   PROC HPMIXED statement, 3829
METHOD= option
   PROC HPMIXED statement, 3829
MMEQ option
   PROC HPMIXED statement, 3829
MODEL statement
   HPMIXED procedure, 3842
NAMELEN= option
   PROC HPMIXED statement, 3829
NLOPTIONS statement
   HPMIXED procedure, 3843
NLPRINT option
   PROC HPMIXED statement, 3829
NOCLPRINT option
   PROC HPMIXED statement, 3829
NOMISS option
   OUTPUT statement (HPMIXED), 3844
NOFIT option
   PROC HPMIXED statement, 3830
NFOULLZ option
   RANDOM statement (HPMIXED), 3850
NOINFO option
   PROC HPMIXED statement, 3830
NOINT option
   MODEL statement (HPMIXED), 3843
NOIT option
   PARMS statement (HPMIXED), 3847
NOITPRINT option
   PROC HPMIXED statement, 3830
ORDER= option
   PROC HPMIXED statement, 3830
OUT= option
   OUTPUT statement (HPMIXED), 3844
PARMS statement
   HPMIXED procedure, 3846
PARMSDATA= option
   PARMS statement (HPMIXED), 3848
PDATA= option
   PARMS statement (HPMIXED), 3848
PDIFF option
   LSMEANS statement (HPMIXED), 3841, 3842
PEARSON= option
   OUTPUT statement (HPMIXED), 3844
PREDICTED= option
   OUTPUT statement (HPMIXED), 3844
PROC HPMIXED statement, see HPMIXED procedure
   HPMIXED procedure, 3826
R option
   REPEATED statement (HPMIXED), 3855
RANDOM statement
   HPMIXED procedure, 3849
RANKS option
   PROC HPMIXED statement, 3831
RC option
   REPEATED statement (HPMIXED), 3855
RCI option
   REPEATED statement (HPMIXED), 3855
RCORR option
   REPEATED statement (HPMIXED), 3855
REF= option
   CLASS statement (HPMIXED), 3832
REPEATED statement
   HPMIXED procedure, 3854, 3879
RESIDUAL= option
   OUTPUT statement (HPMIXED), 3844
RI option
   REPEATED statement (HPMIXED), 3856
SIMPLE option
   PROC HPMIXED statement, 3831
SINGCHOL= option
   PROC HPMIXED statement, 3831
SINGRES= option
   PROC HPMIXED statement, 3831
SINGULAR= option
   CONTRAST statement (HPMIXED), 3835
   ESTIMATE statement (HPMIXED), 3839
   LSMEANS statement (HPMIXED), 3839
   PROC SINGCHOL statement, 3831
   OUTPUT statement (HPMIXED), 3846
   PROC SINGCHOL statement, 3831
   OUTPUT statement (HPMIXED), 3846
   HPMIXED procedure, 3844
   HPMIXED procedure, 3844
SLICE= option
   LSMEANS statement (HPMIXED), 3842
SOLUTION option
   MODEL statement (HPMIXED), 3843
   RANDOM statement (HPMIXED), 3850
STDERR= option
   OUTPUT statement (HPMIXED), 3844
STUDENT= option
   OUTPUT statement (HPMIXED), 3844
SUBJECT= option
   CONTRAST statement (HPMIXED), 3835
   ESTIMATE statement (HPMIXED), 3839
   RANDOM statement (HPMIXED), 3850
   REPEATED statement (HPMIXED), 3856

TEST statement
   HPMIXED procedure, 3856
TRUNCATE option
   CLASS statement (HPMIXED), 3833
TYPE= option
   RANDOM statement (HPMIXED), 3851
   REPEATED statement (HPMIXED), 3856
UCL= option
   OUTPUT statement (HPMIXED), 3844
UPDATE option
   PROC HPMIXED statement, 3831
UPPERB= option
   PARMS statement (HPMIXED), 3849
VARIANCE= option
   OUTPUT statement (HPMIXED), 3844
WEIGHT statement
   HPMIXED procedure, 3857
ZETA= option
   MODEL statement (HPMIXED), 3843