SAS/STAT® 9.22 User's Guide

The HPMIXED Procedure

(Book Excerpt)

SAS® Documentation

This document is an individual chapter from *SAS/STAT® 9.22 User's Guide*.

The correct bibliographic citation for the complete manual is as follows: SAS Institute Inc. 2010. *SAS/STAT® 9.22 User's Guide*. Cary, NC: SAS Institute Inc.

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, May 2010

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/publishing** or call 1-800-727-3228.

Chapter 43

The HPMIXED Procedure

Contents

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 3414 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:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

$$\boldsymbol{\gamma} \sim N(\mathbf{0}, \mathbf{G})$$

$$\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$$

$$\mathrm{Cov}[\boldsymbol{\gamma}, \boldsymbol{\epsilon}] = \mathbf{0}$$

The symbols in these expressions denote the following:

$\mathbf{y}$	the $(n \times 1)$ vector of responses
$\mathbf{X}$	the $(n \times k)$ design matrix for the fixed effects
$\boldsymbol{\beta}$	the $(k \times 1)$ vector of fixed-effects parameters
$\mathbf{Z}$	the $(n \times q)$ design matrix for the random effects
$\boldsymbol{\gamma}$	the $(q \times 1)$ vector of random effects
$\boldsymbol{\epsilon}$	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 $\mathbf{y}$ 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

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

where $\mathbf{G}$ denotes the variance matrix of the random effects. The mixed model crossproduct matrix

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

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 $\mathbf{X}$ matrix, a large number of columns in the $\mathbf{Z}$ 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 $\mathbf{X}$ and/or $\mathbf{Z}$ 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 $\mathbf{X}'\mathbf{X}$ 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 REPEATED statement to model R-side covariance structures
- 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

Getting Started: HPMIXED Procedure

Mixed Model with Large Number of Fixed and Random Effects

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)).

```
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:

```
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 43.1 through Figure 43.4.

The "Class Level Information" table in Figure 43.1 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 $\mathbf{R}$ in the conditional distribution, given the random effects. In the case of the HPMIXED procedure this matrix is simply $\mathbf{R} = \sigma^2\mathbf{I}$ 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 $\mathbf{G}$ matrix is a diagonal (3000×3000) matrix with the common variance on the diagonal.

Figure 43.1 Class Levels and Dimensions

The HPMIXED Procedure

Class Level Information

Class	Levels	Values
Species	5	1 2 3 4 5
Farm	100	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ...
Animal	3000	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ...

Dimensions

G-side Cov. Parameters	1
R-side Cov. Parameters	1
Columns in X	506
Columns in Z	3000
Subjects (Blocks in V)	1

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

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

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 3410.

Figure 43.2 displays the covariance parameter estimates at convergence of the REML algorithm. The variance component estimate for animal effect is $\widehat{\sigma}_a^2 = 3.9889$ and for residual $\widehat{\sigma}^2 = 7.9623$. These estimates are close to the simulated values (4.0 and 8.0).

Figure 43.2 Estimates of Variance Components

Covariance Parameter Estimates

Cov Parm	Estimate
Animal	3.9889
Residual	7.9623

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 43.3 Type III Tests of Fixed Effect

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Species*Farm	495	39500	11.72	<.0001

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 43.4. The small p-value indicates that there are significant differences among the first three genetics species.

Figure 43.4 Result of CONTRAST Statement

Contrasts

Label	Num DF	Den DF	F Value	Pr > F
Species1 = Species2 = Species3	2	39500	92.93	<.0001

Syntax: HPMIXED Procedure

The following statements are available in PROC HPMIXED:

PROC HPMIXED < *options* > ;
BY *variables* ;
CLASS *variables* ;
EFFECT *name* = *effect-type* (*variables* < / *options* >) ;
ID *variables* ;
MODEL *dependent* = < *fixed-effects* > < / *options* > ;
RANDOM *random-effects* < / *options* > ;
PARMS < *(value-list) . . .* > < / *options* > ;
TEST *fixed-effects* < / *options* > ;
CONTRAST *'label' contrast-specification* < , *contrast-specification* > < , . . . > < / *options* > ;
ESTIMATE *'label' contrast-specification* < **(***divisor=n***)** >
< , *'label' contrast-specification* < **(***divisor=n***)** > > < , . . . > < /*options* > ;
LSMEANS *fixed-effects* < / *options* > ;
NLOPTIONS < *options* > ;
OUTPUT < **OUT=***SAS-data-set* >
< *keyword*< **(***keyword-options***)** >< *=name* > >. . .
< *keyword*< **(***keyword-options***)** >< *=name* > > < / *options* > ;
WEIGHT *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, 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."

PROC HPMIXED Statement

PROC HPMIXED < *options* > ;

The PROC HPMIXED statement invokes the procedure. Table 43.1 summarizes important options in the PROC HPMIXED statement by function. These and other options in the PROC HPMIXED statement are then described fully in alphabetical order.

Table 43.1 PROC HPMIXED Statement Options

Option	Description
Basic Options	
DATA=	Specifies input data set
METHOD=	Specifies the estimation method
NOPROFILE	Includes scale parameter in optimization
ORDER=	Determines the sort order of CLASS variables
BLUP	Computes BLUP/BLUE only
Displayed Output	
IC	Displays a table of information criteria
ITDETAILS	Displays estimates and gradients added to "Iteration History"
MAXCLPRINT	Specifies the maximum levels of CLASS variables to print
MMEQ	Displays mixed model equations
NOCLPRINT	Suppresses "Class Level Information" completely or in parts
NOITPRINT	Suppresses "Iteration History" table
SIMPLE	Displays "Descriptive Statistics" table
Singularity Tolerances	
SINGCHOL=	Tunes singularity for Cholesky decompositions
SINGRES=	Tunes singularity for the residual variance
SINGULAR=	Tunes general singularity criterion

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 tolerence 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.

INFOCRIT=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 43.2.

Table 43.2 Information Criteria

Criteria	Formula	Reference
AIC	$-2\ell + 2d$	Akaike (1974)
AICC	$-2\ell + 2dn^*/(n^* - d - 1)$ for $n^* \geq d + 2$ $-2\ell + 2d(d + 2)$ for $n^* < d + 2$	Hurvich and Tsai (1989) and Burnham and Anderson (1998)
HQIC	$-2\ell + 2d \log(\log(n))$	Hannan and Quinn (1979)
BIC	$-2\ell + d \log(n)$	Schwarz (1978)
CAIC	$-2\ell + d(\log(n) + 1)$	Bozdogan (1987)

Here ℓ 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.

The quantities d, n, and 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($\mathbf{X}$).
  - 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($\mathbf{X}$).

- 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.

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=REML

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

$$\begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{Z} \\ \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z} + \widehat{\mathbf{G}}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \\ \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \end{bmatrix}$$

assuming $\widehat{\mathbf{G}}$ is nonsingular. If $\widehat{\mathbf{G}}$ is singular, PROC HPMIXED produces the following coefficients

$$\begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}\widehat{\mathbf{G}} \\ \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}\widehat{\mathbf{G}} + \widehat{\mathbf{G}} \end{bmatrix} \begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \\ \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \end{bmatrix}$$

See the section "Model and Assumptions" on page 3407 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.

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 order in which to sort 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. With this option, the levels of such variables are ordered by their internal value.

The ORDER= option can take the following values:

Value of ORDER=	Levels Sorted By
DATA	Order of appearance in the input data set
FORMATTED	External formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	Descending frequency count; levels with the most observations come first in the order
INTERNAL	Unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine-dependent. For more information about sorting 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*.

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.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC HPMIXED to obtain separate analyses on 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 sorting 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*.

CLASS Statement

CLASS *variables* < / **TRUNCATE** > **;**

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 Language Reference: Dictionary*. You can adjust the order of CLASS variable levels with the ORDER= option in the PROC HPMIXED statement. You can specify the following option in the CLASS statement after a slash (/):

TRUNCATE
specifies that class levels should 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 $\mathbf{L}'\boldsymbol{\phi} = \mathbf{0}$, where $\mathbf{L}' = [\mathbf{K}' \; \mathbf{M}']$ and $\boldsymbol{\phi}' = [\boldsymbol{\beta}' \; \boldsymbol{\gamma}']$, in several inference spaces. The inference space corresponds to the choice of $\mathbf{M}$. When $\mathbf{M} = \mathbf{0}$, 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 **L** matrix is formed. The syntax representation of a *contrast-specification* is
< fixed-effect values ... > < | random-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 **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 3390), 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

$$F = \frac{\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}' \mathbf{L}(\mathbf{L}'\widehat{\mathbf{C}}\mathbf{L})^{-1}\mathbf{L}' \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}}{r}$$

where $r = \text{rank}(\mathbf{L}'\widehat{\mathbf{C}}\mathbf{L})$ and approximates its distribution with an F distribution. In this expression, $\widehat{\mathbf{C}}$ 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 $r = \text{rank}(\mathbf{L}'\widehat{\mathbf{C}}\mathbf{L})$, 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 χ^2 tests be performed in addition to any F tests. A χ^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 χ^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.

E

requests that the **L** matrix coefficients for the contrast 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, 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($\mathbf{K}' - \mathbf{K}'\mathbf{T}$) is greater than *c*number* for any row of $\mathbf{K}'$ in the contrast, then **K** is declared nonestimable. Here **T** is the Hermite form matrix $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$, and *c* is ABS($\mathbf{K}'$) 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.

EFFECT Statement (Experimental)

EFFECT *name = effect-type (variables < / options >)* **;**

The EFFECT statement enables you to construct special collections of columns for design matrices. These collections are referred to as *constructed effects* to distinguish them from the usual model effects formed from continuous or classification variables, as discussed in the section "GLM Parameterization of Classification Variables and Effects" on page 410 of Chapter 19, "Shared Concepts and Topics."

The following *effect-types* are available:

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 43.3 summarizes important options for each type of EFFECT statement.

Table 43.3 Important EFFECT Statement Options

Option	Description
Options for Collection Effects	
DETAILS	Displays the constituents of the collection effect
Options for Lag Effects	
DESIGNROLE=	Names a variable that controls to which lag design an observation is assigned
DETAILS	Displays the lag design of the lag effect
NLAG=	Specifies the number of periods in the lag
PERIOD=	Names the variable that defines the period
WITHIN=	Names the variable or variables that define the group within which each period is defined
Options for Multimember Effects	
NOEFFECT	Specifies that observations with all missing levels for the multimember variables should have zero values in the corresponding design matrix columns
WEIGHT=	Specifies the weight variable for the contributions of each of the classification effects
Options for Polynomial Effects	
DEGREE=	Specifies the degree of the polynomial
MDEGREE=	Specifies the maximum degree of any variable in a term of the polynomial
STANDARDIZE=	Specifies centering and scaling suboptions for the variables that define the polynomial
Options for Spline Effects	
BASIS=	Specifies the type of basis (B-spline basis or truncated power function basis) for the spline expansion
DEGREE=	Specifies the degree of the spline transformation
KNOTMETHOD=	Specifies how to construct the knots for spline effects

For further details about the syntax of these *effect-types* and how columns of constructed effects are computed, see the section "EFFECT Statement (Experimental)" on page 418 of Chapter 19, "Shared Concepts and Topics."

ESTIMATE Statement

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 $\mathbf{L}' = [\mathbf{K}' \; \mathbf{M}']$, as in the CONTRAST statement, where $\mathbf{K}$ is associated with the fixed effects and $\mathbf{M}$ is associated with the G-side random effects.

PROC HPMIXED then produces for each row $\mathbf{l}$ of $\mathbf{L}'$ an approximate t test of the hypothesis $H: \mathbf{l}\boldsymbol{\phi} = 0$, where $\boldsymbol{\phi} = [\boldsymbol{\beta}' \; \boldsymbol{\gamma}']'$. 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 $\mathbf{L}$ matrix for an ESTIMATE statement follows the same rules as listed under the CONTRAST statement.

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 $1 - number$. 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;
```

E

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:

```
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;

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

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 3406).

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.

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 3397 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

$$\mathbf{L}\left(\mathbf{X}'\mathbf{X}\right)^{-}\mathbf{L}'$$

matrix. 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:

```
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.

The differences of the LS-means are displayed in a table titled "Differences of Least Squares Means." For ODS purposes, the table name is "Diffs."

E
requests that the matrix coefficients for all LSMEANS effects be displayed. For ODS purposes, 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 3395.

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

SLICE=*fixed-effect*

SLICE=(*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." For ODS purposes, 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:

```
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, $n - \text{rank}(\mathbf{X})$, 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

$$p = \Pr\{F_{l,\infty} > F_{obs}\} = \Pr\{\chi^2_l > lF_{obs}\}$$

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 3407, the fixed-effects parameter estimates are $\widehat{\boldsymbol{\beta}}$ and their approximate standard errors are the square roots of the diagonal elements of $(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}$.

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.

NLOPTIONS Statement

NLOPTIONS < *options* > **;**

For more information about the NLOPTIONS, see the section "NLOPTIONS Statement" on page 508 in Chapter 19, "Shared Concepts and Topics."

If you choose TECH=NEWRAP, then the default value of LSPRECISION is 0.4 in 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*

DATA=*SAS data set*

specifies the name of the output data set. If the OUT= (or DATA=) 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 43.4 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 43.4 Keywords for Output Statistics

Keyword	Options	Description	Expression	Name
PREDICTED	BLUP	Linear predictor	$\widehat{\eta} = \mathbf{x}'\widehat{\boldsymbol{\beta}} + \mathbf{z}'\widehat{\boldsymbol{\gamma}}$	Pred
	NOBLUP	Marginal linear predictor	$\widehat{\eta}_m = \mathbf{x}'\widehat{\boldsymbol{\beta}}$	PredPA
STDERR	BLUP	Standard deviation of linear predictor	$\sqrt{\mathrm{Var}[\widehat{\eta} - \mathbf{z}'\boldsymbol{\gamma}]}$	StdErr
	NOBLUP	Standard deviation of marginal linear predictor	$\sqrt{\mathrm{Var}[\widehat{\eta}_m]}$	StdErrPA
RESIDUAL	BLUP	Residual	$r = y - \widehat{\eta}$	Resid
	NOBLUP	Marginal residual	$r_m = y - \widehat{\eta}_m$	ResidPA
PEARSON	BLUP	Pearson-type residual	$r/\sqrt{\widehat{\mathrm{Var}}[y\|\boldsymbol{\gamma}]}$	Pearson
	NOBLUP	Marginal Pearson-type residual	$r_m/\sqrt{\widehat{\mathrm{Var}}[y]}$	PearsonPA
STUDENT	BLUP	Studentized residual	$r/\sqrt{\widehat{\mathrm{Var}}[r]}$	Student
	NOBLUP	Studentized marginal residual	$r_m/\sqrt{\widehat{\mathrm{Var}}[r_m]}$	StudentPA
LCL	BLUP	Lower prediction limit for linear predictor		LCL
	NOBLUP	Lower confidence limit for marginal linear predictor		LCLPA
UCL	BLUP	Upper prediction limit for linear predictor		UCL
	NOBLUP	Upper confidence limit for marginal linear predictor		UCLPA
VARIANCE	BLUP	Conditional variance of response variable	$\widehat{\mathrm{Var}}[y\|\boldsymbol{\gamma}]$	Variance
	NOBLUP	Marginal variance of response variable	$\widehat{\mathrm{Var}}[y]$	VariancePA

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 $1 - number$. 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;
   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 6 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–CovP*n*), the residual variance (possibly estimated) for models with a residual variance parameter, and various functions of the likelihood.

For ODS purposes, 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*

PDATA=*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–Covp$q$, 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;
  180 200 170 1000
;
proc hpmixed;
   class A B C rep;
   model yield = A;
   random rep B C;
   parms / pdata=data_covp;
   by year;
run;
```

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 3406). It is, however, assumed that random effects specified using different RANDOM statements are not correlated.

Using notation from the section "Model Assumptions" on page 3407, the purpose of the RANDOM statement is to define the $\mathbf{Z}$ matrix of the mixed model, the random effects in the $\boldsymbol{\gamma}$ vector, and the structure of $\mathbf{G}$. The $\mathbf{Z}$ matrix is constructed exactly like the $\mathbf{X}$ matrix for the fixed effects, and the $\mathbf{G}$ matrix is constructed to correspond to the effects constituting $\mathbf{Z}$. The structure of $\mathbf{G}$ is defined by using the TYPE= option described on page 3406.

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 $1 - number$ be constructed for the predictors of random effects in this statement. The value of *number* must be between 0 and 1 exclusively; the default is 0.05. Specifying the ALPHA= option implies the CL option.

CL

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.

GROUP=*effect*

defines an effect specifying heterogeneity in the covariance structure of $\mathbf{G}$. 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.

NOFULLZ

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

SOLUTION

requests that the solution for the random-effects parameters be produced. Using notation from the section "Model Assumptions" on page 3407, these estimates are the empirical best linear unbiased predictors (BLUPs) $\widehat{\boldsymbol{\gamma}} = \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{V}}^{-1}(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})$. 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 $\widehat{\boldsymbol{\gamma}}$ displayed in the Estimate column; rather, they are the standard errors of predictions $\widehat{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i$, where $\widehat{\boldsymbol{\gamma}}_i$ is the ith BLUP and $\boldsymbol{\gamma}_i$ is the ith random-effect parameter.

SUBJECT=*effect*

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 $\mathbf{G}$ with identical blocks. The $\mathbf{Z}$ 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 covariance structure of $\mathbf{G}$ for G-side effects.

The TYPE=VC (variance components) option is the default structure. Another available structure is CHOL.

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.

E3 | EIII
requests that Type III matrix coefficients be displayed if a Type III test is performed.

CHISQ
requests that χ^2 tests be performed in addition to any F tests. A χ^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 χ^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 $\mathbf{R}$ with $\mathbf{W}^{-1/2}\mathbf{R}\mathbf{W}^{-1/2}$, where $\mathbf{W}$ 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.

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), and Brown and Prescott (1999).

The HPMIXED procedure fits models generally of the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

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

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 $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$. For both procedures a key assumption is that $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$ are normally distributed with

$$\mathrm{E}\begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$
$$\mathrm{Var}\begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix}$$

The two procedures differ in their assumptions about the variance matrices $\mathbf{G}$ and $\mathbf{R}$ for $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$, respectively. The MIXED procedure allows a variety of different structures for both $\mathbf{G}$ and $\mathbf{R}$; while in HPMIXED procedure, $\mathbf{R}$ is always assumed to be of the form $\mathbf{R} = \mathbf{I}\sigma^2$, and the structures available for modeling $\mathbf{G}$ 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*:

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

Let $\mathbf{C}$ denote the coefficient matrix of the mixed model equations:

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

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

$$L(\boldsymbol{\theta}, \sigma^2) = -2 \log l = (n - p) \log(2\pi) + \log |\mathbf{C}| + \log |\mathbf{G}| + n \log(\sigma^2) + \mathbf{y}'\mathbf{P}\mathbf{y}$$

where

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

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

Computing and Maximizing the Likelihood

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

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

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

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

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

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

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 θ_i and θ_j are

$$\frac{\partial L}{\partial \theta_i} = \mathrm{tr}\left(\frac{\partial \mathbf{V}}{\partial \theta_i}\mathbf{P}\right) - \mathbf{y}'\mathbf{P}\frac{\partial \mathbf{V}}{\partial \theta_i}\mathbf{P}\mathbf{y}$$

and

$$\frac{\partial^2 L}{\partial \theta_i \theta_j} = -\mathrm{tr}\left(\frac{\partial \mathbf{V}}{\partial \theta_i}\mathbf{P}\frac{\partial \mathbf{V}}{\partial \theta_j}\mathbf{P}\right) + 2\mathbf{y}'\mathbf{P}\frac{\partial \mathbf{V}}{\partial \theta_i}\mathbf{P}\frac{\partial \mathbf{V}}{\partial \theta_j}\mathbf{P}\mathbf{y}$$

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 $\mathbf{G} = \mathbf{I}\sigma_a$, the first derivative of the log likelihood with respect to the parameter σ_a^2 can be computed as follows:

$$\frac{\partial L}{\partial \sigma_a^2} = \frac{q}{\sigma_a^2} - \frac{\text{tr}(\mathbf{C}^{aa})}{\sigma_a^4} - \frac{\widehat{\boldsymbol{\gamma}}'\widehat{\boldsymbol{\gamma}}}{\sigma_a^4}$$

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

The second derivative of the log likelihood needs to be computed only if you specify certain nondefault optimization techniques in the NLOPTIONS statement, namely TECH=NEWRAP, TECH=NRRIDG, or TECH=TRUREG; see "NLOPTIONS Statement" on page 508 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

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

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 $\mathbf{y}$. This quadratic expression defines the average information (Johnson and Thompson 1995) with respect to θ_i and θ_j:

$$\text{AI}(\theta_i, \theta_j) = \mathbf{y}'\mathbf{P}\frac{\partial \mathbf{V}}{\partial \theta_i}\mathbf{P}\frac{\partial \mathbf{V}}{\partial \theta_j}\mathbf{P}\mathbf{y}$$

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 $\mathbf{y}$ with the unobservable random effects $\boldsymbol{\gamma}$, leading to a simplified form for the log likelihood. For example, if $\mathbf{G} = \mathbf{I}\sigma_a^2$ then given the realized values $\tilde{\boldsymbol{\gamma}}$ of the unobservable random effects $\boldsymbol{\gamma}$, the REML estimate of σ_a^2 satisfies

$$\widehat{\sigma}_a^2 = \frac{\tilde{\boldsymbol{\gamma}}'\tilde{\boldsymbol{\gamma}}}{q - \sigma^2/\sigma_a^2 \text{tr}(\mathbf{C}^{aa})}$$

This corresponds to the maximization step of EM-REML. However, the true realized values $\tilde{\boldsymbol{\gamma}}$ are unknown in practice. The expectation step of EM-REML replaces them with the conditional expected values $\widehat{\boldsymbol{\gamma}}$ of the random effects, given the observed data $\mathbf{y}$ and initial values for the parameters. The new estimate of σ_a^2 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 3407. 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 $\mathbf{C}$ be the matrix of mixed model equations of size 5×5.

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

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

i	1	1	2	2	3	4	5
j	1	4	2	3	3	4	5
C_{ij}	8.0	2.0	4.0	3.0	5.0	7.0	9.0

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 nonzero elements gets very large. The classical and widely used scheme, and the one the HPMIXED procedure employs, is the (ic, jc, 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 jc; that is, if $c(k) = C_{ij}$, then $jc(k) = j$. To identify the individual rows, you need to know where each row starts and ends. These row starting positions are stored in the vector ic. For instance, if C_{ij} is the first nonzero element in the row i and $c(k) = C_{ij}$, then $ic(i) = k$. The row i ending position is one

less than $ic(i+1)$. Thus, the number of nonzero elements in the row i is $ic(i+1) - ic(i)$, these elements in the row i are stored consecutively starting from the position $k_i = ic(i)$

$$c(k_i), c(k_i+1), c(k_i+2), ..., c(k_{i+1}-1)$$

and the corresponding columns indices are stored consecutively in

$$jc(k_i), jc(k_i+1), jc(k_i+2), ..., jc(k_{i+1}-1)$$

For example, the seven nonzero elements in matrix $\mathbf{C}$ are stored in (ic, jc, c) format as

ic	1	3	5	6	7	8	
jc	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 (ic, jc, 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 3378 has 3506 mixed model equations. Storing just the upper triangle of these equations in a dense form requires $(1 + 3506) \times 3506/2 = 6,147,771$ 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 $\mathbf{C}$ and then forms the $\mathbf{L}$ coefficient matrix to obtain the F value as follows:

$$F = \frac{\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}' \mathbf{L}'(\mathbf{L}\widehat{\mathbf{C}}^{-1}\mathbf{L}')^{-1}\mathbf{L} \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}}{r}$$

where $r = \text{rank}(\mathbf{L}\widehat{\mathbf{C}}^{-1}\mathbf{L}')$. The coefficient matrix $\mathbf{L}$ 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 HPMIXED 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: HPMIXED Procedure" on page 3378.

Default Output

The following sections describe the output PROC HPMIXED 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.

For ODS purposes, 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 HPMIXED statement. For ODS purposes, 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. For ODS purposes, 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 and for each covariate in the MODEL statement.

Iteration History

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

For ODS purposes, the name of the "Iteration History" table is "IterHistory."

Covariance Parameter Estimates

The "Covariance Parameter Estimates" table contains the estimates of the parameters in $\mathbf{G}$ and $\mathbf{R}$. 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.

For ODS purposes, 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. For ODS purposes, 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 3383.

For ODS purposes, 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 43.5.

Table 43.5 ODS Tables Produced by PROC HPMIXED

Table Name	Description	Required Statement / Option
OverallANOVA	ANOVA table for model without random effect	Default output for fixed models
ClassLevels	Level information from the CLASS statement	Default output
Coef	**L** matrix coefficients	E option in MODEL, CONTRAST, ESTIMATE, or LSMEANS
Contrasts	Results from the CONTRAST statements	CONTRAST
ConvergenceStatus	Convergence status	Default
CovParms	Estimated covariance parameters	Default output
Diffs	Differences of LS-means	LSMEANS / DIFF (or PDIFF)
Dimensions	Dimensions of the model	Default output
Estimates	Results from ESTIMATE statements	ESTIMATE
FitStatistics	Fit statistics	Default
IterHistory	Iteration history	Default output
LSMeans	LS-means	LSMEANS
MMEq	Mixed model equations	PROC HPMIXED MMEQ
ModelInfo	Model information	Default output
NObs	Number of observations read and used	Default output
OptInfo	Optimization information	Default output
ParameterEstimates	Fixed-effects solution	MODEL / SOLUTION
ParmSearch	Parameter search values	PARMS
SimpleStatistics	Descriptive statistics for dependent variable and covariate variables	PROC HPMIXED SIMPLE
Slices	Tests of LS-means slices	LSMEANS / SLICE=
SolutionR	Random-effect solution vector	RANDOM / SOLUTION
Tests3	Type III tests of fixed effects	TEST

Examples: HPMIXED Procedure

Example 43.1: 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:

```
%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.

```
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;
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 43.1.1 displays the EBVs of the top 10 animals, along with their precision and confidence bounds.

Output 43.1.1 Estimated Breeding Values: Top 10 Animals

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

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.

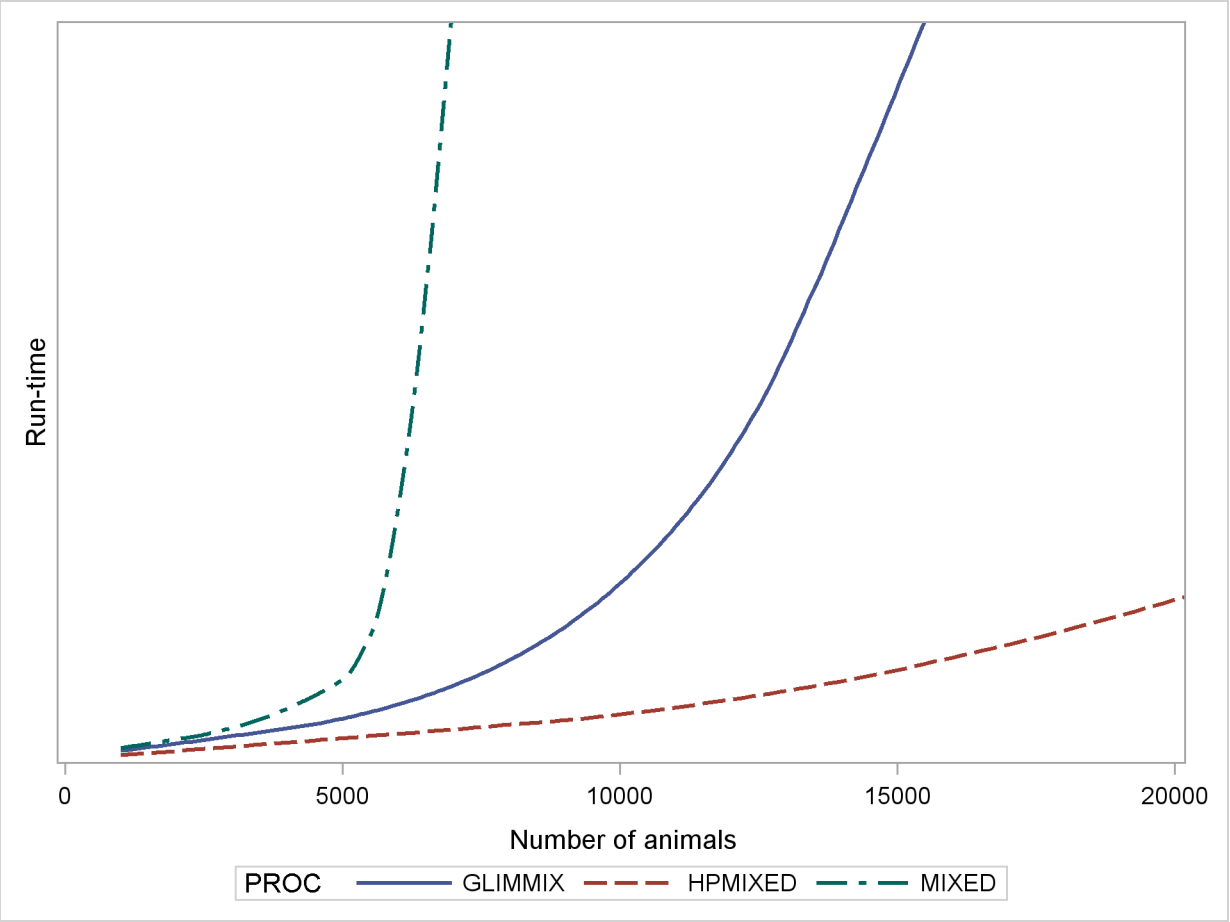
You can also use PROC MIXED and PROC GLIMMIX to compute EBVs, but the performance of these general mixed modeling procedures for this specialized kind of data and model is quite different from that of PROC HPMIXED. The MIXED and GLIMMIX procedures are engineered to have good performance properties across 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 43.1.2 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.

Output 43.1.2 Comparing Mixed Model Tools for Increasingly Sparse Problems

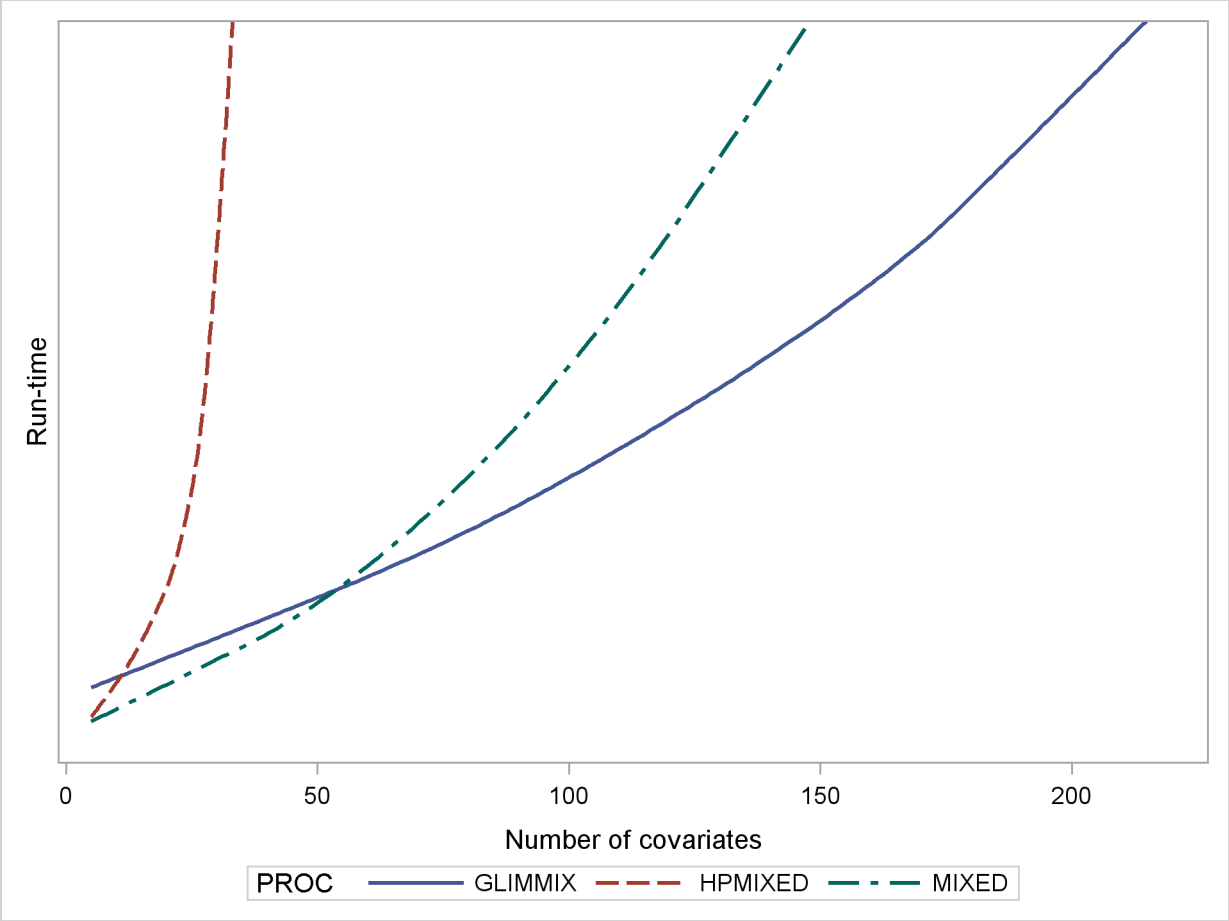


The vertical axis in Output 43.1.2 measures run time, but the units are omitted: relative performance is what counts, and that is expected to be fairly invariant to machine architecture. The output shows that while the performance of the MIXED and GLIMMIX procedures is relatively competitive with PROC 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 43.1.3 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.

Output 43.1.3 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 43.2: Comparing Results from PROC HPMIXED and PROC MIXED

This example revisits the mixed model problem from the section "Getting Started: MIXED Procedure" on page 4518, in Chapter 56, "The MIXED Procedure," with the data set shown in the following statements:

```
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 67   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
;
```

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:

```
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 43.2.1. The MIXED procedure converges after six iterations and achieves a –2 restricted log likelihood of 71.02246.

Output 43.2.1 Iteration History and Fit Statistics: MIXED Procedure

The Mixed Procedure

Iteration History

Iteration	Evaluations	-2 Res Log Like	Criterion
0	1	74.11074833	
1	2	71.51614003	0.01441208
2	1	71.13845990	0.00412226
3	1	71.03613556	0.00058188
4	1	71.02281757	0.00001689
5	1	71.02245904	0.00000002
6	1	71.02245869	0.00000000

Fit Statistics

-2 Res Log Likelihood	71.0
AIC (smaller is better)	77.0
AICC (smaller is better)	79.0
BIC (smaller is better)	75.2

Output 43.2.2 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 $\mathbf{X}'\mathbf{X}$ 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 $\mathbf{X}'\mathbf{X}$ matrix is a linear function of previous columns. Consequently, the coefficient for the second level of the Gender variable is zero.

Output 43.2.2 Parameter Estimates and Solutions: MIXED Procedure

Covariance Parameter Estimates

Cov Parm	Estimate
Family	2.4010
Family*Gender	1.7657
Residual	2.1668

Solution for Fixed Effects

Effect	Gender	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		68.2114	1.1477	3	59.43	<.0001
Gender	F	−3.3621	1.1923	3	−2.82	0.0667
Gender	M	0	.	.	.	.

Solution for Random Effects

Effect	Gender	Family	Estimate	Std Err Pred	DF	t Value	Pr > |t|
Family		1	1.2680	1.1201	10	1.13	0.2840
Family		2	0.08980	1.1121	10	0.08	0.9372
Family		3	−1.6660	1.1712	10	−1.42	0.1853
Family		4	0.3082	1.1201	10	0.28	0.7888
Family*Gender	F	1	−0.3198	1.0810	10	−0.30	0.7734
Family*Gender	M	1	1.2523	1.0933	10	1.15	0.2787
Family*Gender	F	2	−0.4299	1.0774	10	−0.40	0.6983
Family*Gender	M	2	0.4959	1.0774	10	0.46	0.6551
Family*Gender	F	3	−0.08229	1.1409	10	−0.07	0.9439
Family*Gender	M	3	−1.1429	1.1409	10	−1.00	0.3401
Family*Gender	F	4	0.8320	1.0933	10	0.76	0.4642
Family*Gender	M	4	−0.6053	1.0810	10	−0.56	0.5878

The "Type 3 Tests of Fixed Effects" table in Output 43.2.3 is produced by the MIXED procedure by default.

Output 43.2.3 Test of Gender Effect

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Gender	1	3	7.95	0.0667

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

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

Output 43.2.4 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 43.2.1).

Output 43.2.4 Iteration History and Fit Statistics: HPMIXED Procedure

The HPMIXED Procedure

Iteration History

Iteration	Evaluations	Objective Function	Change	Max Gradient
0	4	71.023177956	.	0.034074
1	3	71.022519936	0.00065802	0.007839
2	3	71.022477283	0.00004265	0.004674
3	2	71.0224587	0.00001858	0.000168
4	2	71.022458689	0.00000001	3.28E-6

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 43.2.5 displays the results that correspond to those in Output 43.2.2 in the MIXED procedure.

Output 43.2.5 Parameter Estimates and Solutions: HPMIXED Procedure

Covariance Parameter Estimates

Cov Parm	Estimate
Family	2.4010
Family*Gender	1.7657
Residual	2.1668

Solution for Fixed Effects

Effect	Gender	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		0	.	.	.	.
Gender	F	64.8493	1.1477	16	56.50	<.0001
Gender	M	68.2114	1.1477	16	59.43	<.0001

Solution for Random Effects

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

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 50 and "Linear Model Theory" on page 59, 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 43.2.5 and Output 43.2.2 are identical. Also, the test for the Gender effect yields the same F value in both analyses (compare Output 43.2.6 and Output 43.2.3).

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 use 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 43.2.6 Parameter Estimates and Solutions: HPMIXED Procedure

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Gender	1	16	7.95	0.0123

Example 43.3: 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 3378 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:

```
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.

```
proc hpmixed data=Sim;
   class Species Farm Animal;
   model Yield = Farm|Species;
   random Animal;
   test Species Species*Farm;
   ods output CovParms=HPMEstimate;
run;

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 43.3.1 and Output 43.3.2. 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 43.3.1 Iteration History for the HPMIXED Procedure

The HPMIXED Procedure

Iteration History

Iteration	Evaluations	Objective Function	Change	Max Gradient
0	4	202516.66891	.	0.841954
1	6	202516.66887	0.00004385	0.000641
2	1	202516.66887	-0.00000000	0.000641

Output 43.3.2 Iteration History for the GLIMMIX Procedure

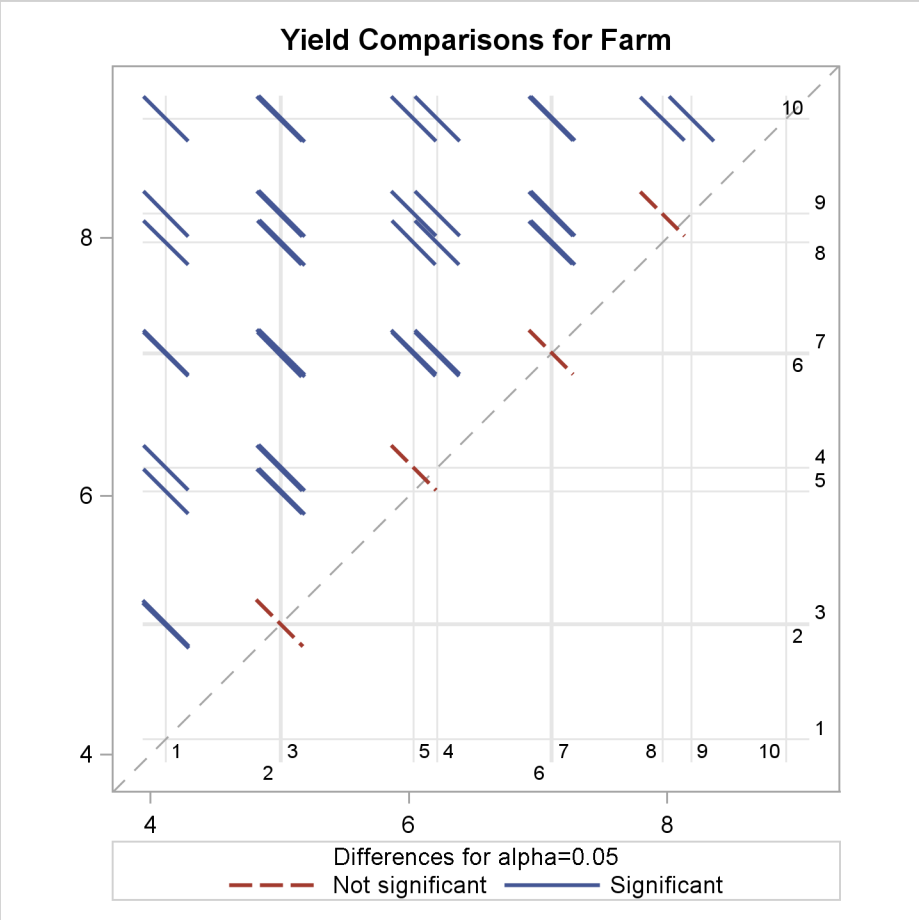
The GLIMMIX Procedure

Iteration History

Iteration	Restarts	Evaluations	Objective Function	Change	Max Gradient
0	0	4	202516.66887	.	0

The graphical multiple-comparisons analysis for the LS-means of farms is shown in Output 43.3.3. It confirms the pairwise equalities between farm effects with which the data were simulated.

Output 43.3.3 LS-Means Plot of Pairwise Farm Differences



For more information about the interpretation of the LS-means difference plot, see the section "ODS Graphics" on page 2837, in Chapter 38, "The GLIMMIX Procedure."

Example 43.4: 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.

```
%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;
   end;

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;
       end;
    run;
```

A linear mixed model for fitting the log intensity data Y_{ijkmnr} from such a design is described by Littell et al. (2006) as follows:

Y_{ijkmnr}	$=$		**Fixed Effects**
		μ	Overall mean
	$+$	λ_i	Gene
	$+$	τ_j	Treatment
	$+$	δ_k	Dye
	$+$	$(\tau\lambda)_{ij}$	Treatment-by-gene
	$+$	$(\delta\lambda)_{ik}$	Dye-by-gene
	$+$	p_r	Pin
			Random Effects
	$+$	a_m	Microarray
	$+$	$(a\lambda)_{im}$	Microarray-by-gene
	$+$	$d(a)_{mn}$	Dip-within-mircroarray
	$+$	$(ap)_{mr}$	Microarray-by-pin
	$+$	e_{ijkmnr}	Residual noise

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

```
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 43.4.1 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 43.4.1 Mixed Model Dimensions

The HPMIXED Procedure

Dimensions

G-side Cov. Parameters	4
R-side Cov. Parameters	1
Columns in X	4513
Columns in Z	3054
Subjects (Blocks in V)	1

The *p*-value in Output 43.4.2 indicates that there are significant differences between treatments.

Output 43.4.2 Type III Tests of Fixed Effects

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Trt	5	20497	370005	<.0001

References

Akaike, H. (1974), "A New Look at Statistical Model Identification," *IEEE Transactions on Automatic Control*, 19, 716–723.

Bozdogan, H. (1987), "Model Selection and Akaike's Information Criterion (AIC): The General Theory and Its Analytical Extensions," *Psychometrika*, 52, 345–370.

Brown, H. and Prescott, R. (1999), *Applied Mixed Models in Medicine*, New York: John Wiley & Sons.

Burnham, K. P. and Anderson, D. R. (1998), *Model Selection and Inference: A Practical Information-Theoretic Approach*, New York: Springer-Verlag.

Churchill, G. A. (2002), "Fundamentals of Experimental Design for cDNA Microarray," *Nature Genetics*, 32, 490–495.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society, Series B*, 39, 1–38.

George, J. A. and Liu, J. W. (1981), *Computer Solutions of Large Sparse Positive Definite Systems*, Englewood Cliffs, NJ: Prentice-Hall.

Gibson, G. and Wolfinger, R. D. (2004), "Gene Expression Profiling Using Mixed Models," in A. M. Saxton, ed., *Genetic Analysis of Complex Traits Using SAS*, 251–278, Cary, NC: SAS Publishing.

Gilmour, A. R., Thompson, R., and Cullis, B. R. (1995), "Average Information REML: An Efficient Algorithm for Variance Parameter Estimation in Linear Mixed Models," *Biometrics*, 51, 1440–1450.

Hannan, E. J. and Quinn, B. G. (1979), "The Determination of the Order of an Autoregression," *Journal of the Royal Statistical Society, Series B*, 41, 190–195.

Henderson, C. R. (1990), "Statistical Method in Animal Improvement: Historical Overview," in *Advances in Statistical Methods for Genetic Improvement of Livestock*, 1–14, New York: Springer-Verlag.

Hurvich, C. M. and Tsai, C.-L. (1989), "Regression and Time Series Model Selection in Small Samples," *Biometrika*, 76, 297–307.

Johnson, D. L. and Thompson, R. (1995), "Restricted Maximum Likelihood Estimation of Variance Components for Univariate Animal Models Using Sparse Matrix Techniques and Average Information," *Journal of Dairy Science*, 78, 449–456.

Kerr, M. K., Martin, M., and Churchill, G. A. (2000), "Analysis of Variance for Gene Expression Microarray Data," *Journal of Computational Biology*, 7, 819–837.

Littell, R. C., Milliken, G. A., Stroup, W. W., and Wolfinger, R. D. (1996), *SAS System for Mixed Models*, Cary, NC: SAS Institute Inc.

Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., and Schabenberger, O. (2006), *SAS for Mixed Models*, Second Edition, Cary, NC: SAS Press.

McLean, R. A., Sanders, W. L., and Stroup, W. W. (1991), "A Unified Approach to Mixed Linear Models," *The American Statistician*, 45, 54–64.

Ott, E. R. (1967), "Analysis of Means—A Graphical Procedure," *Industrial Quality Control*, 24, 101–109. Reprinted in *Journal of Quality Technology*, 15 (1983), 10–18.

Schabenberger, O., Gregoire, T. G., and Kong, F. (2000), "Collections of Simple Effects and Their Relationship to Main Effects and Interactions in Factorials," *The American Statistician*, 54, 210–214.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Searle, S. R., Casella, G., and McCulloch, C. E. (1992), *Variance Components*, New York: John Wiley & Sons.

Shewchuk, J. R. (1994), *An Introduction to the Conjugate Gradient Method without the Agonizing Pain*, Technical report, Carnegie Mellon University, Pittsburgh, PA.

Tsuruta, S., Misztal, I., and Stranden, I. (2001), "Use of the Preconditioned Conjugate Gradient Algorithm as a Generic Solver for Mixed-Model Equations in Animal Breeding Apllications," *Journal of Animal Science*, 79, 1166–1172.

Verbeke, G. and Molenberghs, G., eds. (1997), *Linear Mixed Models in Practice: A SAS-Oriented Approach*, New York: Springer.

Verbeke, G. and Molenberghs, G. (2000), *Linear Mixed Models for Longitudinal Data*, New York: Springer.

Winer, B. J. (1971), *Statistical Principles in Experimental Design*, Second Edition, New York: McGraw-Hill.

Wolfinger, R. D., Gibson, G., Wolfinger, E., Bennett, L., Hamadeh, H., Bushel, P., Afshari, C., and Paules, R. S. (2001), "Assessing Gene Significance from cDNA Microarray Expression Data via Mixed Models," *Journal of Computational Biology*, 8, 625–637.

Subject Index

Syntax Index

Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to **yourturn@sas.com**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **suggest@sas.com**.