GENEKIT

another BLUP software
(Vincent Ducrocq, June 30, 2011)

1. Background

GENEKIT is a BLUP software, initially derived from the *blupf90* program of Ignazy Mizstal, with which it has very little in common now. Initially, it was implemented to improve some aspects of *blupf90*, in particular:

- A more user-friendly parameter file, where variables have names (and not just order numbers, source of many mistakes);
- A more efficient solution algorithm: a preconditioned conjugate gradient algorithm in which the preconditioner is not just a diagonal but an incomplete block Cholesky factor of the coefficient matrix. This proved to be very efficient to quickly move information in one iteration from progeny to parents several generations away and back, with a strong effect on convergence rate;
- A more efficient (but more restricted) approach for multiple trait settings, with systematic canonical decomposition including in situations with missing values on some traits and/or heterogeneous residual variances.

Other features were (and still are being) included, depending on personal needs. These include:

- an approximation of reliabilities using Harris and Johnson's (1998. *J. Dairy Sci.*, 81:2723-2728) approach (not available for all models);
- Reliabilities extended to the random regression case (Ducrocq and Schneider., 2007. Generalization of the information source method to compute reliabilities in test day models. *Interbull Bulletin* n°37, 82-87)
- a modelling of heterogeneous residual variances;
- an easier treatment of univariate (fixed and random) regressions with storage of continuous covariates (splines, Legendre polynomials, eigenvectors, etc.) in small tables (no need to have all coefficients for each record in the input data file);
- an automatic production of some by-product results (residuals, mendelian sampling estimates, pre-adjusted records, daughter deviations, equivalent daughter matrices and vectors of daughter yield deviation for test day models, etc.);
- some tools to monitor convergence, restarts and storage.

2. Installation

The version considered here is the one available on June 30, 2011. All the subroutines and the makefile for unix and linux systems and a test example are in the file genekit20110630.tar.Z. To use it, first copy to an appropriate directory, uncompress and untar. Modify the Makefile options according to your fortran compiler. Compile.
To use the program, use the script called ***genekit*.** If *param* is your parameter file, just type:

genekit param or *genekit param > log_filename*

At INRA, you can find GENEKIT in the /logiciels/GENEKIT directory for the AIX (unix) system and in /ugen/ugenvpd/bao/GENEKIT on the linux machine dga8 and dga10 (and soon for dga11)

Under GENEKIT, you will find the *genekit* script (which can be copied in your directory) in the BIN directory, all elements to run the example in EX/test_genekit.tar, this manual in the MAN directory and the source code in the SRC directory (all programs and subroutine in *genekit_300611.tar*, or in uncompressed form in *last_version*/). You will also find the executable *genekit.exe* but you don't necessary need to copy it if you use the *genekit* script. Note: for old GENEKIT users using AIX(unix), previous versions of the executable can be reached by simply adding OLD/ after BIN.

3. Distribution

GENEKIT is for your own use, at your own risk. Please do not distribute it. Any request from new users should go through me. Among other things, I want to keep track of who is using GENEKIT to be able to inform users about bugs and new versions.

4. Parameter file

List of main keywords

Any line starting with **#** is a comment

In bold underlined : compulsory keywords (not underlined = optional keywords)

TITLE
<u>**DATAFILE**</u>
OLD_CODE_FILE
STORAGE
NCOMBIMAX
SYSTEM_SIZE
<u>**SOLUTIONS**</u>
CHECK_CONVERGENCE
STARTING_VALUES
<u>**COLUMN_NAMES**</u>
TABLES
ALIAS
<u>**NAME_OF_TRAITS**</u>
<u>**MODEL**</u>
MODEL_HETEROGENEITY

```
                                              ^
      RANDOM_HETEROGENEITY                    |
      RANDOM_TYPE_HETEROGENEITY   (as many times as necessary)
      VARIANCE_HETEROGENEITY
      AUTOCORRELATION_PARAMETER               v
```

<u>**TYPE_OF_EFFECTS**</u>
WEIGHTS
ALGORITHM
CONVERGENCE
MAX_ITERATIONS
BLOCK_CHOLESKY
FILL_IN
DEFINE_COMBINED_SOLUTIONS
TREATED_AS_MISSING
COMPUTE_RELIABILITY
BY_PRODUCT_OUTPUT
DEFINE_BY_PRODUCTS
BY_PRODUCT_COLUMNS
<u>**RANDOM_RESIDUAL**</u>

```
                                       ^
   RANDOM_GROUP                        |
   RANDOM_TYPE             (as many times as necessary)
   FILE                                |
   (CO)VARIANCES                       v
```

<u>**END**</u>

5. Detailed description of keywords

TITLE

The following line is used as a title in the output.

<u>**DATAFILE**</u>

(compulsory) The next line is the name of the input data file. This file (as any other file used or written by GENEKIT) is an ASCII, free format file will blanks used as separators

OLD_CODE_FILE

(Only useful at INRA, for those who used the *recode* program of Bernard Bonaïti for preparation of the input data file)

The solution file(s) will include the original code (can be a combination of codes). The name of the file connecting old to new code (created by recode) must be given on the next line.

STORAGE

If "**STORAGE ON_DISK**", the data file will be repeatedly read on disk. Be aware that this option may have a quite detrimental impact on CPU time

Default value: **STORAGE IN_CORE**. the data file is read and stored in core memory.

SYSTEM_SIZE

On the next line: upper bound *u* of the number of nonzero elements of the mixed model coefficient matrix.

If not specified, the default value is *u* = max(100000, *coef* * number of equations) where *neff* is the number of fixed and *random* effects and *coef* = max(6,1+*neff**(*neff*+1)/2)).

Be careful: if a very large number is specified, you may ask too much and GENEKIT may crash because of lack of memory… Conversely, if you put a value which is unnecessarily too large, GENEKIT will tell you. A trial and error approach should lead to an optimum value (large enough, without crashing)

If you later use **MODEL_HETEROGENEITY** to model the logarithm of residual variances, you may need a second value *v* after *u* (on the same line): *v* is an upper bound of the number of nonzero elements in the linear system used to estimate the effects on the residual variance. The default value is *v*=100000 and is usually enough.

NCOMBIMAX

The next line indicates the maximum number of combinations of missing records in a multiple trait setting (for example, with 3 traits, this number is 7 (no missing traits, trait 1, 2, 3 or 1 and 2, 1 and 3, 2 and 3 may be missing – 1, 2 and 3 missing is not counted as it is .. a missing record). The default value is large (400) so this statement may be needed only in very large multiple trait evaluations.

<u>**SOLUTIONS**</u>

(compulsory) on the next line: name (or root of names) of the solution files(s)

Only one solution file is created, except if after the solution file, one adds the "**separate"** keyword:

"**SOLUTIONS separate**".

Then one solution file is created for each effect in the model with name= root name followed by "*.eff*" if *eff* is the name of the effect given in COLUMN_NAMES. An integer can also be added after the file name (or after "**combined**" or "**separate**", if these keyworks are used) to indicate the frequency at which solutions are stored. For
example, "solfile 40" or "solfile separate 40" implies that the file will be stored every 40 iterations and specific convergence criteria (changes between solutions 40 iterations apart) will be computed.

STARTING_VALUES

On the next line: name of the file with starting values (for example, solutions obtained after a reduced number of iterations).
The file(s) has (have) the same structure as the solution file(s). For example, one may have one file per effect in the model. In this case, indicate:

"STARTING_VALUES separate"

CHECK_CONVERGENCE

On the next line: name of the file with solutions that will be used to compute extra convergence criteria (for example, distance to "exact" solutions, if the exact solutions were previously stored). This may be used to test different implementation strategies to achieve faster convergence.
The file(s) has (have) the same structure as the solution file(s). For example, one may have one file per effect in the model. In this case, use:

"CHECK_CONVERGENCE separate"

<u>COLUMN_NAMES</u>

(compulsory) On the next lines, (ordered) names of the variables in the data file. Any variable used as cross-classified effect in the model should be coded from 1 to *k*, where *k* is the number of levels for that effect.

TABLES

On the next lines, one can specify continuous variables that the program will create for fixed or random regression models.
Currently implemented are spline coefficients, Legendre polynomials coefficients, and coefficients read from an external file.

- Example with splines:
  c1 c2 c3 c4 = **splines[**DCC**]** splin1 100 150 200 265

creates a table called "*splin1*" with all spline coefficients here called c1, c2, c3, c4 corresponding to cubic splines with 4 knots at 100, 150, 200 and 265. These coefficients will be stored in core and will be referred to through the variable DCC.
In other words, only DCC is needed in the input data file: c1,c2, c3, c4 are automatically created for all records.
In case **splines_no_int** is used instead of **splines**, the first spline coefficient (intercept) is not used (case where splines are nested within a particular effect).

- Example with Legendre polynomials
  d1 d2 d3 **legendre[**DIM**]** leg 5 335

creates a table called "leg" for the coefficients (d1, d2, d3) of a Legendre polynomial of order 2 between 5 and 335. These coefficients will be stored in core and will be referred to through the variable DIM.
In other words, only DIM is needed in the input data file. d1, d2, d3 are automatically created for all records.

- Example with external files
  vp1 vp2 vp3 vp4 = **external_file[**DIM**]** TD/tab_vpl_G

This will read the coefficients vp1 to vp4 (for example, coefficients associated with 4 eigenvectors) from file TD/tab_vpl_G . They will be stored in core and will be referred to through the variable DIM.
In other words, only DIM is needed in the input data file. vp1, vp2, vp3, vp4 are automatically created for all records.

One can have several splines or Legendre polynomials or external files defined on different lines (with a different set of variable names c1,c2… and a different table name). A separate file is created for each table name.

ALIAS

In some cases, one may want to use the same variable(s) twice, for example *year* may be fit directly into the linear model but also in the model of log- residual variances. To prevent confusion while avoiding storing twice the same information in the input data file, one can use aliases.

On the next line(s), a second name is given to each variable of interest. For example:

year2 = year

This new name can later be used as if it was actually present in the list of variables of the input data file.

NAME_OF_TRAITS

(compulsory) The next line lists the names of the traits analysed. This is necessarily a subset of the list given in "COLUMN_NAMES"

MODEL

(compulsory) The next line(s) describes the model of analysis. Examples:

- *trait1 = herd age year* (=simple fixed effects model)
- *trait1 trait2 = herd age[year]* (bivariate analysis age nested with year)
  or equivalently : *trait1 = herd age[year]*
  trait2 = herd age[year]
- *trait1 = herd d1 d2 d3* (spline or Legendre or any coefficients defined in TABLE)
- *trait1 = herd d1[age] d2[age] d3[age]* (polynomials or splines within age class)
- *trait1 = herd age animal*
  trait2= herd age animal
  trait3= herd age animal (multiple trait animal model)
- *trait1 = herd dcc c1[age] c2[age] c3[age] c4[age] d1[anim] d2[anim] d3[anim]*
  (fixed and random regression model)

A number of restrictions exist. Some may disappear in the future with new versions of *genekit*. The main ones are <u>that models for multiple trait analyses must be identical and must include no more than one random effect other than the residual (but missing data are allowed</u>).

MODEL_HETEROGENEITY

Specifies that the model has heterogeneous residual variances. If the second keyword is **multiplicative** as in:

MODEL_HETEROGENEITY MULTIPLICATIVE

Heterogeneous residuals variances affect all effects (fixed et random effects: multiplicative model)

otherwise only the random part is affected as in Robert-Granié et al (1999). Accounting for variance heterogeneity in French dairy cattle genetic evaluation, *Livestock Production Science*, 60, 343-357.

The next line(s) describes the model for the logarithm of the residual variance, when needed. Examples:

- *size = age year2*
  which means that the expected value of ln(residual variance) is the sum of an *age* effect and a *year* effect.
- *size udder feet = age year2 technician*
- *Milk = region herdyear*

RANDOM_HETEROGENEITY

The next line lists the effect in the model of the log-residual variance to be treated as random (e.g., *technician* or h*erdyear* in the last two models above). If several effects are treated as random, this statement and the following two (or three) are repeated as often as needed.

RANDOM_TYPE_HETEROGENEITY

The next line defines the variance structure of the effect appearing in RANDOM_HETEROGENEITY. Two possibilities are offered:

- **diagonal**
  The levels of the *technician* effects are uncorrelated
- **autocorrelation** *filename*
  The levels of the *herdyear* effects are autocorrelated. To specify the structure (*herdyear* effects are autocorrelated within herd), the file *filename* specifies this structure: it has (at least) 3 columns: column 1 has consecutive numbers (1 to nhy_max), column 2 has herd effects, columns 3 has year effect. This file must be sorted appropriately: within a herd, if two consecutive lines have year number equal to 3 and 4 respectively, the correlation between the corresponding *herdyear* effects is the autocorrelation ρ (also works correctly for years more than one year apart).
  Example:
  ```
  1 1 1
  2 1 2
  3 2 4
  4 2 5
  5 2 7
  6 2 8 …
  ```

VARIANCE_HETEROGENEITY

The next line specifies the value of the variance of the effect appearing in RANDOM_ HETEROGENEITY. If *n* traits are analysed, *n* values should appear.
If VARIANCE_HETEROGENEITY is followed by **FIXED** (on the same line), the values indicated will remain constant; otherwise they will be iteratively updated.

AUTOCORRELATION_PARAMETER

The next line specifies the value of the autocorrelation for the effect appearing in RANDOM_TYPE_HETEROGENEITY with autocorrelation.
If AUTOCORRELATION_PARAMETER is followed by **FIXED** (on the same line), the value indicated will remain constant; otherwise it will be iteratively updated.

TYPE_OF_EFFECTS

(compulsory) The next line(s) indicate(s) the type of effect in the model(s) (including for the of the log-residual variance).
One line per type (***cross*** = cross-classified, **cov** = continuous covariate). Example:

cross herd age anim
cov dcc c1 c2 c3 c4 d1 d2 d3

Cross-classified effects must have been recoded from 1 to *k*, where *k* is the number of levels
There is no need to specify the type of nested effects (e.g., d1[age] or age[herd] for a different continuous covariate d1 for each age level or a age by herd effect)

WEIGHTS

The next line gives variable name(s) which represent(s) the weight associated to the records of each trait. The names point to either a variable of the input data file (a variable included in the COLUMN_NAMES list) or a name defined in the TABLE or ALIAS statements. For multiple traits, each weight name must be connected to the relevant trait by adding "[*name_of_the_trait*]".
For example: *sizw[sire] udw[udder]*
When no weight name is attached to a particular trait, the default value is a weight of 1 for all records for that trait. When no weight is attached to any trait, either delete the statement or leave a blank line after WEIGHTS.

ALGORITHM

The next line specifies the algorithm to be used for solution of the linear system. Currently, three possibilities are given:

direct, **conjugate_gradient** or **conjugate_gradient on_data**

- **direct** leads to the exact solution using *fspak* subroutines. It is applicable only in the univariate case. This is the default value (this may be changed in the future).
- **conjugate_gradient**. The solution relies upon a preconditioned conjugate gradient (PCG) algorithm of univariate linear systems (after canonical transformation in case of multivariate analyses), where the preconditioner -only one, even for multiple trait analyses- is an incomplete Cholesky factorization of the coefficient matrix. The keywords **FILL_IN** and **BLOCK_CHOLESKY** give flexibility to the choice of the preconditioner (see below).
- **conjugate_gradient on_data** is for very large application when the full coefficient matrix cannot/should not be stored in core memory. Then, calculations are based iteration on data techniques.

CONVERGENCE

The next line includes the value of the desired convergence criterion. If residual variance heterogeneity is accounted for, a second criterion is needed on the same line for the log-residual variance model. The convergence criterion is (currently) the average absolute change in solutions between two consecutive iterations. The default values are 0.0001 and 0.001. Note that with PCG, this is not necessarily the best convergence criterion: it is often quite conservative.

Adding the keyword **BASED_ON_RANDOM** leads to slightly more reliable checks of convergence, based only on random effects: as no constraint on fixed effects is imposed, complex cases may lead to “drifts” of fixed effects (e.g., one varying by Δ between two iterations while another is varying by $-\Delta$) even after random effects have converged.

The statement is ignored when the algorithm chosen is ***direct***.

MAX_ITERATIONS

The next line includes one, two or three numbers specifying the maximum of iterations performed before the program is stopped..

The first one is for regular mixed models solution. The second is relevant only for multiple trait analyses with missing values on some traits and indicates the number of conjugate gradient iterations to perform between two updates of the missing values (a value less than 5 is recommended; a value of 1 is usually fine). The third value specifies the maximum number of updates of the effects describing the heterogeneous residual variances.

For example, “40 1 15” means that 15 times (at most, i.e. if convergence criteria are not reached before) 40 conjugate gradient iterations will be performed before new estimates (possibly together with the variance and/or autocorrelation of random effects) of the model describing the heterogeneous residual variances are computed.

If a value of 0 is specified for the third parameter (e.g.,”200 1 0”) and the program is started reading previous solutions, the estimates of the heterogeneous residual variances are used and kept fixed during the whole run.

The default values are 200, 1 and 20.

The statement is ignored when the algorithm chosen is ***direct***.

BLOCK_CHOLESKY

The preconditioner of the PCG algorithm is an incomplete Cholesky factor (ICF) of (univariate) mixed model coefficient matrix. Note that an “exact” ICF, that is a Cholesky factor for which *all* the positions where nonzero elements in the initial coefficient matrix are kept in the ICF is too time-consuming and above all, fails in many cases (requires the square root of a negative number). Then, the computation of the preconditioner is restarted after the addition of a constant positive value on the diagonal of the coefficient matrix. If the decomposition fails again, the constant is doubled and so on, until the decomposition can be performed. But in such cases, the final preconditioner can be very inefficient and convergence can be very slow.

It was found that a sparser ICF, where the decomposition is performed on blocks of the coefficient matrix, is faster, requires less storage and is much more efficient than the exact ICF. The BLOCK_CHOLESKY statement permits to choose the appropriate blocks. The default value (that is when BLOCK_CHOLESKY is not specified) is one block per effect. Then the preconditioner is very sparse (= a diagonal, except for the blocks involving a relationship matrix). But in some cases, other blocking strategies may be better. For example, it seems that all spline coefficients for one particular effect may be advantageously treated together in a single block. This is done by simply putting on a same line all effects that should be treated in the same block. Examples:

herd
dcc
c1 c2 c3
d1 d2 d3 d4

or

herd dcc c1 c2 c3 d1 d2 d3 d4 (= exact ICF)

or

herd
…
d3 (= default)
d4

FILL_IN

This is another way to improve the preconditioner. The next line indicates the number *m* of extra terms added to each line of the ICF (whatever the blocking strategy), i.e., *m* elements which were 0 in the initial coefficient matrix are allowed to be nonzero in the ICD. These elements are chosen by the preconditioning subroutine. The larger *m*, the better the ICF. For a very large *m*, the ICF tends to the exact Cholesky factor. But an increase of *m* automatically leads to a slower factorisation and to more memory space required.

The default value is 0. Larger values than 20 are not recommended.

TREATED_AS_MISSING

The next line gives the rules used to define missing values. These are similar to the ones used in PEST. Three numbers are needed. All values smaller than the first one, equal to the second one or larger than the third one are considered as missing.

For example: *-100. 0. 50.* means that values less than -100, equal to 0 or larger than 50 will be considered as missing.

Do not forget this statement if 0 is considered as a missing value!

DEFINE_COMBINED_SOLUTIONS

The next lines specify linear combinations of the original estimated breeding values in random regression models, representing for example the average lactation production, the production in first lactation, a measure of persistency, etc.

Each line is composed of a new variable name, the '=' sign, a coefficient and the name of an effect appearing in the MODEL statement and if needed and as many times as necessary, a '+' sign, another coefficient and another name of an effect. For example:

g_all = 0.5 vg1 + 0.3 vg2 + 0.2 vg3

A specific solution file will be created named "*solutionfilename_comb*"

COMPUTE_RELIABILITY

The next lines lead to the computations of specific by-products of genetic evaluations: specifies: <u>scalar reliabilities</u> using Harris and Johnson's information source approach (Harris and Johnson, 1998, Approximate reliabilities of genetic evaluations under an animal model. *J. Dairy Sci. 81,* 2723-2728), matrix reliabilities (Ducrocq and Schneider., 2007. Generalization of the information source method to compute reliabilities in test models. Bulletin n°37, 82-87) for random regression models, daughter yield deviations and corresponding ECD for test-day models (Tauebert et al, 2010. An approach to compute EDC and DYD for test-day models. 9WCGALP, Leipzig, Germany, 231).

If the first line following COMPUTE_RELIABILITY is **yes** or **yes_only**, the reliability is computed and stored in a file derived from the solution file name by adding *".reliab"*. In case of "*yes_only*", the program stops after the reliability computation, before iterating on mixed model equations. If the line is blank, reliability will not be computed.
If **yes** is not followed by **tdm** (on the same line), the model is not a random regression model (e.g., a test-day model). Then several options are available:

- If *yes* or *yes_only* is followed by **ignore_correlations**, reliabilities will be univariate ones, even though a multiple trait model has been defined.
- If *yes* or *yes_only* is followed by **without_grand_children**, reliabilities will be based only on parent, own performance and first generation progeny. *ignore_correlations* and *without_grand_children* can be used together.

If **yes** is followed by **tdm**, the model is not a random regression model (e.g., a test-day model). Again, a number of options are proposed:

- *tdm* can be followed by **dyd** (on the same line), in which case a vector of daughter yield deviations (DYD, or more correctly, daughter corrected performances, i.e., average performances of the daughters of a male corrected for all non genetic effects in the model and the additive genetic contribution of their dam) and a matrix of equivalent daughter contributions (EDC) are computed.
  In fact these are transformed in such a way that they appear as several pseudo-records which can be used as new records for a random regression model (several scalar DYD and corresponding vectors of EDC coefficients are created in a file called "*solutionfilename.dyd*" which can be used as an imput file. Contact me for details.
- The next line may contain the keyword **dyd_combined**, in which case DYD and scalar EDC are computed for the combined effects defined in DEFINE_COMBINED_SOLUTIONS.

Whether the model is a random regression model, better reliability estimates are obtained if the following statements are added on the next lines:

- **contemporary** *cg_eff* where *cg_eff* represents a "contemporary group" effect, for example, a herd effect, which will be absorbed in order to account in the reliability calculation for the loss of information due to its estimation.
- **permanent** *pe_eff* or:
- **permanent** *pe_eff1 pe_eff2 pe_eff"* … for random regression models
  pe_eff or the *pe_eff_i* represent the permanent environment effect(s) which are to be absorbed too.

BY_PRODUCT_OUTPUT

The next line specifies the file name where by-product information will be stored.

DEFINE_BY_PRODUCTS

This statement allows the definition of new variables to be included in the by-product file and which cannot be simply specified with the keywords available to compute standard outputs (see below).

- These new variables can be functions of the data and the estimates of some effects can be obtained using the keywords **calculate** or **calculatew**
  One line for each new variable describes how to compute it as:
  New_variable = calculate[variable coef1 eff1 coef2 eff2 …]
  For example, the statements:

corrected = calculate[milk -1 hys -1 lact -1 age_at_calving]
sum_fixed= calculate[1 hys   1 lact 1 age_at_calving]

will define the new variable *corrected* after taking away the estimates of the herd-year-season, lactation and at calving effects and the variable *sum_fixed* will compute the sum of these three fixed effects from each milk yield record in the original dataset.

The *corrected* and *sum_fixed* variables must appear in the BY_PRODUCT_COLUMNS statement.

If *calculatew* is used in models where heterogeneous residual variances are modelled, the new variables are standardized to a common residual variance.

- **new_weight** is used when the initial weight (specified in WEIGHTS or 1) is corrected to take into account the contemporary group size (i.e., if the contemporary group size is n, the new weight is the initial one multiplied by (1 – 1/n)
  corr_weight =new_weight[milk] is the weightof the trait "*milk*" accounting for the *cg_eff* contemporary group effect indicated in "*contemporary cg_eff*" of the COMPUTE_RELIABILITY statement. *corr_eff* will appear for each initial record if it is specified in the list of BY_PRODUCT_COLUMNS below.

BY_PRODUCT_COLUMNS

The next line indicates the names of all the variables that will be stored in the by product file. These can be any variables coming from the original file as well as:

- *Residuals* (records corrected for all estimated fixed and random effects). They are indicated by **RESID[***trait_name***]** or by **RESIDW[***trait_name***]** if the residuals are to be standardized to a common residual variance (corresponding to a weight of 1)
- *Pre-ajusted records* (records corrected for all estimated fixed and random effects, except for the random effects to which a relationship matrix is attached). They are indicated by **PREADJ[***trait_name***]** or by **PREADJW[***trait_name***]** if the residuals are to be standardized to a common residual variance (corresponding to a weight of 1).
- *"daughter deviations" (*records corrected for all estimated fixed and random effects, except for the random effects to which a relationship matrix is attached, but including the contribution of the dam). They are indicated by **D_DEV[***trait_name***]** or by **D_DEVW[***trait_name***]** if the residuals are to be standardized to a common residual variance (corresponding to a weight of 1).
  Note that this statement is not valid for random regression models.
- *mendelian sampling terms* (computed as the genetic effect of an animal (or a sum of all effects to which a relationship matrix is attached) minus half the sum of the genetic effect of its parents. They are indicated by **MENDEL[***trait_name***]**.
- *estimated weights* of residuals in models accounting for heterogeneous residual variance. This is indicated by **WR[***trait_name***]**.
  In all cases, missing values are indicated by -9999. This is essential to remember that when some of these values (e.g., pre-adjusted records) are to be used in another evaluation, TREATED_AS_MISSING has to be modified accordingly.

RANDOM_RESIDUAL

(compulsory) The next lines specify the residual variance matrix (or scalar = one line) to be used. Each line of the matrix must be on one line with any format but with blanks between columns. If a univariate analysis is desired for each trait, whatever the form of the (co)variance matrix, write **RANDOM_RESIDUAL univariate**. This is the default when a model accounting for heterogeneous residual variances is applied on several traits in the same analysis.

RANDOM_GROUP

The next line specifies the name of one or several effects treated as random with a given (co)variance pattern, for example: "*animal*" for an animal model or "d1 d2 d3" for a random regression model. This keyword as well as the next ones (**RANDOM_TYPE, FILE** and **(CO)VARIANCES**) are repeated as often as needed.

RANDOM_TYPE

The next line indicates the type of (co)variance structure used. It can be either

- **diagonal**
- **add_animal** *(regular relationship matrix, 0 = unknown parent)*
- **add_an_upg** *(relationship matrix with unknown parent groups; unknown parents group codes are larger than animal codes)*
- **add_an_upgi** (*same as add_an_upg but with inbreeding coefficients column)*
- **add_sire** *(relationship matrix between males (sires-maternal grandsire))*
- **add_an_upg** *(relationship matrix between sires-maternal grandsires with unknown parent groups; unknown parents group codes are larger than animal codes)*

FILE

The next line is either blank (**diagonal** *case*) or contains the name of the pedigree file. This pedigree file has three columns (animal, sire, dam or animal, sire, maternal grand-sire) except for **add_an_upgi** for which a fourth column specifies the inbreeding coefficient. Animals are numbered from 1 to n_a; unknown parent groups from n_a+1 to n_a+n_{upg}

(CO)VARIANCES

The next lines specify the random effect(s) (co)variance matrix (or scalar = one line) to be used. Each line of the matrix must be on one line with any format but with blanks between columns.

END

All lines after this statement are ignored.

6. Examples

- **A univariate mixed model**

```
TITLE
Example from Helene
DATAFILE
/ugend/ugenhel/TD/datavinc.dat
#OLD_CODE_FILE            <= commented out
#newcode                  <= commented out
SYSTEM_SIZE
400000
SOLUTIONS
true
STARTING_VALUES
                           <= don't forget the blank line (or delete STATING_VALUES)
COLUMN_NAMES
ani_p ani_g hy dtar HTD calv_m camp DIM calv_a numlac dim2 DCC Fat coef_weightl
eig1 eig2 eig3 eig4 eig5 eig6 tvp1 tvp2 tvp3 tvp4 tvp5 tvp6
NAME_OF_TRAITS
Fat
MODEL
Fat = HTD calv_m calv_a tvp1[ani_g]  tvp2[ani_g] tvp3[ani_g] tvp4[ani_g]
tvp5[ani_g] tvp6[ani_g]
TYPE_OF_EFFECT
cross HTD calv_m calv_a
cov  tvp1 tvp2 tvp3 tvp4 tvp5 tvp6
WEIGHTS
coef_weightl
ALGORITHM
conjugate_gradient
CONVERGENCE
0.00001
MAX_ITERATIONS
500
FILL_IN
2
TREATED_AS_MISSING
-9999 0 9999
RANDOM_RESIDUAL
4200000
RANDOM_GROUP
tvp1 tvp2 tvp3 tvp4 tvp5 tvp6
RANDOM_TYPE
add_an_upg
FILE
/ugend/ugenhel/TD/ped.dat
(CO)VARIANCES
7680453.2     0.0000    0.0000   0.000   0.0000   0.0000
  0.0000    938010.9    0.0000   0.000   0.0000   0.0000
  …
```

- **The same example with 50 iterations more and use of tables**

```
TITLE
Second example from Helene
DATAFILE
/ugend/ugenhel/TD/datavinc.dat
SYSTEM_SIZE
400000
SOLUTIONS
true2
```

```
STARTING_VALUES
true
COLUMN_NAMES
ani_p ani_g hy dtar HTD calv_m camp DIM calv_a numlac dim2 DCC Fat coef_weightl
eig1 eig2 eig3 eig4 eig5 eig6 tvp1 tvp2 tvp3 tvp4 tvp5 tvp6
TABLES
coef = external_file [dim2] tab_weight
vp1 vp2 vp3 vp4 vp5 vp6 = splines[dim2] splin_coef   5 20 50 135 245 335
# could also be, if splin_coef already exists :
#vp1 vp2 vp3 vp4 vp5 vp6 = external_file [dim2] splin_coef
# note: tvp1 to tvp6 and coef_weightl are no longer needed in the data file
NAME_OF_TRAITS
Fat
MODEL
Fat = HTD calv_m calv_a vp1[ani_g]  vp2[ani_g] vp3[ani_g] vp4[ani_g] vp5[ani_g]
vp6[ani_g]
TYPE_OF_EFFECT
cross HTD calv_m calv_a dim2
cov  tvp1 tvp2 tvp3 tvp4 tvp5 tvp6
WEIGHTS
coef
ALGORITHM
conjugate_gradient on_data
CONVERGENCE
0.00001
MAX_ITERATIONS
50
FILL_IN
5
TREATED_AS_MISSING
-9999 0 9999
RANDOM_RESIDUAL
4200000
RANDOM_GROUP
vp1 vp2 vp3 vp4 vp5 vp6
RANDOM_TYPE
add_an_upg
FILE
/ugend/ugenhel/TD/ped.dat
(CO)VARIANCES
      7680453.2    0.0000      0.0000    0.000     0.0000     0.0000
     0.0000       938010.9     0.0000    0.000     0.0000     0.0000
    …
```

- **A multiple trait example**

```
TITLE
Type evaluation: 3 traits example
DATAFILE
type.dat
SOLUTIONS
solu  separate
STARTING_VALUES

COLUMN_NAMES
vis age_an sta_an age_ty sta_ty poinan ps point anim elev  date na nb ireg ian cp1f
pseu VTRA PSIL DPLJ
NAME_OF_TRAITS
VTRA PSIL DPLJ
MODEL
VTRA=age_an sta_an vis anim
PSIL=age_an sta_an vis anim
DPLJ=age_an sta_an vis anim
TYPE_OF_EFFECT
cross vis  anim age_an sta_an
WEIGHTS

ALGORITHM
conjugate_gradient
CONVERGENCE
0.0001
MAX_ITERATIONS
200 2
BLOCK_CHOLESKY
vis
anim
age_an
sta_an
FILL_IN
0
TREATED_AS_MISSING
-10000 0 10000
RANDOM_RESIDUAL
   .500219   .000000   .000000
   .000000  1.666590   .181359
   .000000   .181359   .450611
RANDOM_GROUP
anim
RANDOM_TYPE
add_an_upg
FILE
pedig.recod
(CO)VARIANCES
   .124748   .000000   .000000
   .000000   .497880   .082468
   .000000   .082468   .248692
```

- **Same example with modelling of residual variance heterogeneity and generation of pre-adjusted records**

```
TITLE
Type evaluation: 3 traits example, with heterogeneous residual variance
DATAFILE
type.dat
SOLUTIONS
solu2  separate
```

```
STARTING_VALUES
solu
COLUMN_NAMES
vis age_an sta_an age_ty sta_ty poinan ps point anim elev  date na nb ireg ian cp1f
pseu VTRA PSIL DPLJ
NAME_OF_TRAITS
VTRA PSIL DPLJ
MODEL
VTRA PSIL DPLJ=age_an sta_an vis anim
MODEL_HETEROGENEITY
VTRA=age_ty sta_ty poinan ps
PSIL=age_ty sta_ty poinan ps
DPLJ=age_ty sta_ty poinan ps
RANDOM_HETEROGENEITY
poinan
RANDOM_TYPE_HETEROGENEITY
diagonal
VARIANCE_HETEROGENEITY
0.05 0.05 0.05
TYPE_OF_EFFECT
cross vis  anim age_an sta_an poinan age_ty sta_ty ps
#WEIGHTS
#
ALGORITHM
conjugate_gradient
CONVERGENCE
0.0001 0.005
MAX_ITERATIONS
40 1 12
#BLOCK_CHOLESKY    <=  could be ignored because = default values
#vis
#anim
#age_an
#sta_an
#FILL_IN
#0
TREATED_AS_MISSING
-10000 0 10000
BY_PRODUCT_OUTPUT FILE
preadj.dat
BY_PRODUCT_COLUMNS
age_an sta_an age_ty sta_ty poinan elev point na nb ireg ian ntypepo cp1f ps
PREADJW[VTRA] PREADJW[PSIL] PREADJW[DPLJ] anim
RANDOM_RESIDUAL UNIVARIATE
   .500219   .000000   .000000
   .000000  1.666590   .181359
   .000000   .181359   .450611
RANDOM_GROUP
anim
RANDOM_TYPE
add_an_upg
FILE
pedig.recod
(CO)VARIANCES
   .124748   .000000   .000000
   .000000   .497880   .082468
   .000000   .082468   .248692
```

- **An example for total merit index calculation**

```
TITLE
Total merit index with 11 traits
DATAFILE
data_isu
SOLUTIONS
isutot separate
STARTING_VALUES

COLUMN_NAMES
ani mu miscod lait tp cell long EPTA ATAV DPLJ EQUI HATA VTRA FERG FERV W_lait W_tp
W_cell W_long W_mo W_FERG W_FERV fail
NAME_OF_TRAITS
lait tp cell long EPTA ATAV DPLJ EQUI HATA VTRA FERV
MODEL
 lait tp cell long EPTA ATAV DPLJ EQUI HATA VTRA FERV = mu ani
TYPE_OF_EFFECT
cross ani mu
WEIGHTS
W_lait[lait] W_tp[tp] W_cell[cell] W_long[long] W_FERV[FERV]
ALGORITHM
conjugate_gradient
CONVERGENCE
0.00005
MAX_ITERATIONS
300
TREATED_AS_MISSING
-10000000. -9999.  10000000.
COMPUTE_RELIABILITY
yes
RANDOM_RESIDUAL
    .9590E+06   .0000E+00  -.1645E+06  -78.34 .0000E+00   .0000E+00 ....
       ...
RANDOM_GROUP
ani
RANDOM_TYPE
add_an_upg
FILE
ped_isu
(CO)VARIANCES
    .5755E+06  -512.1   …
  …
```

- **A full test day model with computations of reliabilities and by-products**

```
TITLE
      Full test day model from Helene Leclerc
DATAFILE
      Fix_3
SYSTEM_SIZE
      600000000 50000000
SOLUTIONS
      Rand 50      <=  same name as starting value file ➔ the original file will be overwritten
STARTING_VALUES
      Rand
COLUMN_NAMES
      ani_p HTD HTDnl cmvel cavel ctar HY mvel_nl tar_nl agev_nl reg_nl
      reg_camp tyqula ani DIM DCC nlac nlacb camp TP precor RESIDW
TABLES
      coef_w = external_file[DIM] TP/tab_weight
      vg1 vg2 vg3 vg4 = external_file[DIM] TP/GU
      vp1 vp2 vp3 vp4 = external_file[DIM] TP/PU
      vh1 vh2 = external_file[DIM] TP/HU
NAME_OF_TRAITS
      precor
MODEL
      precor =   HTD cmvel cavel ctar vg1[ani] vg2[ani] vg3[ani] vg4[ani]
vp1[ani_p] vp2[ani_p] vp3[ani_p] vp4[ani_p] vh1[HY] vh2[HY]
MODEL_HETEROGENEITY
      TP =  reg_camp reg_nl tyqula HY
RANDOM_HETEROGENEITY
      HY
RANDOM_TYPE_HETEROGENEITY
      autocorrelation R66/HY
VARIANCE_HETEROGENEITY
      0.100
AUTOCORRELATION_PARAMETER
      0.542
TYPE_OF_EFFECT
      cross  HTD cmvel cavel ctar
      cov    vg1 vg2 vg3 vg4 vp1 vp2 vp3 vp4 vh1 vh2
WEIGHTS
      coef_w
ALGORITHM
      conjugate_gradient on_data
CONVERGENCE
      0.0001
MAX_ITERATIONS
      50 1 6
BLOCK_CHOLESKY
      HTD
      cmvel
      cavel
      ctar
      vg1
      vg2
      vg3
      vg4
      vp1
      vp2
      vp3
      vp4
      vh1
      vh2
FILL_IN
```

```
      3
TREATED_AS_MISSING
      -9999 0 9999
COMPUTE_RELIABILITY
      yes tdm dyd
      dyd_combined
      contemporary HTD
      permanent vp1 vp2 vp3 vp4
DEFINE_COMBINED_SOLUTIONS
      G1 =  -80.403  vg1 + -47.879  vg2 +  -109.949  vg3 +   28.343  vg4
      G2 =  -87.691  vg1 + -43.526  vg2 +   102.239  vg3 +   47.495  vg4
      G3 =  -88.784  vg1 + -42.483  vg2 +    -3.373  vg3 +  -71.567  vg4
      P1 =   78.737  vp1 +  35.369  vp2 +    58.900  vp3 +  115.308  vp4
      P2  =  97.880  vp1 +  39.424  vp2 +    64.690  vp3 +  -97.969  vp4
      P3  =  87.155  vp1 +  45.906  vp2 +  -125.567  vp3 +    1.384  vp4
BY_PRODUCT_OUTPUT FILE
      Rand
BY_PRODUCT_COLUMNS
      ani_p HTD HTDnl cmvel cavel ctar HY mvel_nl tar_nl agev_nl reg_nl
      reg_camp tyqula ani DIM DCC nlac nlacb camp TP precor      RESIDW[precor]
      RESID[precor]
RANDOM_RESIDUAL VALUES
      100
RANDOM_GROUP
      vg1 vg2 vg3 vg4
RANDOM_TYPE
      add_an_upg
FILE
      ped3L [ column=4 A14 A14
(CO)VARIANCES RANDOM_UPG
      32.802859710502   0   0   0
      0   2.90080352973564   0   0
      0   0   0.860172329925959   0
      0   0   0   0.344246117676763
RANDOM_GROUP
      vp1 vp2 vp3 vp4
RANDOM_TYPE
      diagonal
FILE

(CO)VARIANCES
      9.02241492205739   0   0   0
      0   4.20965859092144   0   0
      0   0   1.72523104239631   0
      0   0   0   1.37666480149357
RANDOM_GROUP
      vh1 vh2
RANDOM_TYPE
      diagonal
FILE

(CO)VARIANCES
      16.1561544277188   0
      0   11.5271786876526
END
```

7. The test example

- **Description**

The example is a simulated test day (random regression) model evaluation used to validate the software as in Leclerc et al (2008). The data set was generated based on a real data structure but the performances were constructed based on simulated (i.e., known) fixed and random effects including the residual, in such a way that the exact BLUP solutions are the simulated ones. For details see: Leclerc H., Wensch-Dorendorf M., Wensch J, Ducrocq V and Swalve H.H., 2008 A general method to validate breeding value prediction software. *J. Dairy Sci.* 91: 3179-3183.
The whole files are in the tar file *test_genekit.tar*
The data set *datsim_pet3L.dat* and the pedigree *pedsim_pet3L.dat* files are in the TDM_test directory.
The model for the trait "Milk" is a test day model including fixed effects for herd-testday (HTD), calving month (calv_m), calving age (calv_age) and length of dry period (tar) as well as fixed regressions curves on days carried calf (DCC) nested within lactation, days in milk (DIMr) within calving month and lactation, within length of dry period and lactation, random regression curves on days in milk for 4 additive genetic effects (vg1 to vg4, after transformation to make them uncorrelated), 4 permanent environment effects (vp1 to vp4) and 6 Herd-year effects. Residuals are weighted (weight = coef_w)..
The fixed and random coefficients (legendre polynomials, splines, and specific coeffiocients) stored in different tables in the TDM_test directory
The parameter file leads to the BLUE and BLUP estimation of all fixed and random effects as well as the computation of estimated residuals which are stored in a *byprod_new* file.
To run
The solutions can be compared with the exact solutions (sotred in the TDM_test directory), for example using the sas program *anal_pet3L.sas* which is supplied.

- **Parameter file** (named pet3L.par)

```
TITLE
Montbéliarde TDM data with 3 simulated lactations s
DATAFILE
TDM_test/datsim_pet3L.dat
#STORAGE
#on_disk
SYSTEM_SIZE
500000
SOLUTIONS
sol_pet3L 40
#STARTING_VALUES
#sol_pet3L
COLUMN_NAMES
ani_p HTD calv_m calv_a tar Hy DIMr mvel_nl tar_nl anim DCC nlac camp Milk
TABLES
coef_w = external_file[DIMr] TDM_test/tab_weight_pet3L
#51a 52a 53a 54a = splines_no_int[DCC] splin1b 100 150 200 265
vg1 vg2 vg3 vg4 = external_file[DIMr] TDM_test/tab_legG_pet3L
vp1 vp2 vp3 vp4 = external_file[DIMr] TDM_test/tab_legP_pet3L
vh1 vh2 vh3 vh4 vh5 vh6 = external_file[DIMr] TDM_test/tab_legH_pet3L
51a 52a 53a 54a = external_file[DCC] TDM_test/tab_spline_DCCpet3L
71a 72a 73a 74a 75a 76a = external_file[DIMr] TDM_test/tab_spline_DIMpet3L
81a 82a 83a 84a 85a 86a = external_file[DIMr] TDM_test/tab_spline_DIMpet3Lb
#ALIAS
#dim2=dim
NAME_OF_TRAITS
Milk
MODEL
Milk =  HTD calv_m calv_a tar 51a[nlac] 52a[nlac] 53a[nlac] 54a[nlac] 71a[mvel_nl]
72a[mvel_nl] 73a[mvel_nl] 74a[mvel_nl] 75a[mvel_nl] 76a[mvel_nl] 81a[tar_nl]
82a[tar_nl] 83a[tar_nl] 84a[tar_nl] 85a[tar_nl] 86a[tar_nl] vg1[anim] vg2[anim]
vg3[anim] vg4[anim] vp1[ani_p] vp2[ani_p] vp3[ani_p] vp4[ani_p] vh1[Hy] vh2[Hy]
```

```
vh3[Hy] vh4[Hy] vh5[Hy] vh6[Hy]
TYPE_OF_EFFECT
cross   HTD calv_m calv_a tar
cov     51a 52a 53a 54a 71a 72a 73a 74a 75a 76a 81a 82a 83a 84a 85a 86a vg1 vg2 vg3
vg4 vp1 vp2 vp3 vp4 vh1 vh2 vh3 vh4 vh5 vh6
WEIGHT(S)
coef_w
ALGORITHM
conjugate_gradient
#conjugate_gradient on_data
#direct
CONVERGENCE BASED_ON_RANDOM
0.0001
MAX_ITERATIONS
500
BLOCK_CHOLESKY
HTD
calv_m
calv_a
tar
51a 52a 53a 54a
71a 72a 73a 74a 75a 76a
81a 82a 83a 84a 85a 86a
vg1 vg2 vg3 vg4 vp1 vp2 vp3 vp4
vh1 vh2 vh3 vh4 vh5 vh6
FILL_IN
0
TREATED_AS_MISSING
-999999 0 999999
BY_PRODUCT_OUTPUT FILE
 byprod_new
BY_PRODUCT_COLUMNS
 ani_p HTD DIMr Milk RESID[Milk] RESIDW[Milk]
RANDOM_RESIDUAL VALUES
250
RANDOM_GROUP
vg1 vg2 vg3 vg4
RANDOM_TYPE
add_an_upg
FILE
TDM_test/pedsim_pet3L.dat
(CO)VARIANCES
    3.368877    0.000000    0.000000    0.000000
    0.000000    1.770808    0.000000    0.000000
    0.000000    0.000000    0.577778    0.000000
    0.000000    0.000000    0.000000    0.184885
RANDOM_GROUP
vp1 vp2 vp3 vp4
RANDOM_TYPE
diagonal
FILE

(CO)VARIANCES
     2.365029    0.000000    0.000000    0.000000
     0.000000    1.799412    0.000000    0.000000
     0.000000    0.000000    0.592118    0.000000
     0.000000    0.000000    0.000000    0.496845
RANDOM_RESIDUAL VALUES
250
RANDOM_GROUP
vh1 vh2 vh3 vh4 vh5 vh6
```

```
RANDOM_TYPE
diagonal
FILE

(CO)VARIANCES
     4.329102    0.000000    0.000000    0.000000    0.000000    0.000000
     0.000000    1.510365    0.000000    0.000000    0.000000    0.000000
     0.000000    0.000000    1.307681    0.000000    0.000000    0.000000
     0.000000    0.000000    0.000000    0.799326    0.000000    0.000000
     0.000000    0.000000    0.000000    0.000000    0.591790    0.000000
     0.000000    0.000000    0.000000    0.000000    0.000000    0.432
END
```

- **Commented output file** (named *out*)
  Note: here only the important information is described. A large number of the lines are skipped, indicated below by **"………….."**

Start with the title, the date of analysis, and a copy of the parameter file (with some extra information)

……….

```
         ***********************************
            Montbéliarde TDM data with 3 simulated lactations s
         ***********************************

    GENEKIT version June 30, 2011 -  date of analysis: Wed Jun 29 16:36:41 2011
```

Describe the data file and its contents

```
FILES:
 Parameter file:              pet3L.par
 Data file:                   TDM_test/datsim_pet3L.dat
 Solution file:               sol_pet3L
     stored every  40 iterations

CONTENT OF FILES:
 Number of columns           14
 Column names:               ani_p  HTD  calv_m  calv_a  tar  Hy  DIMr  mvel_nl  tar_nl  anim
                             DCC  nlac  camp  Milk  coef_w  vg1  vg2  vg3  vg4  vp1
                             vp2  vp3  vp4  vh1  vh2  vh3  vh4  vh5  vh6  51a
                             52a  53a  54a  71a  72a  73a  74a  75a  76a  81a
                             82a  83a  84a  85a  86a
```

Describe the variables read in tables

```
 Number of dummy variables created 31
      created in the TABLES section :
 ----
 in table: TDM_test/tab_weight_pet3L, indexed by variable DIMr
    Column    coef_w  = value obtained from column  1 of external file

 first 5 lines and last 5 lines  of table TDM_test/tab_weight_pet3L
   1   0.352179
   2   0.358633
   3   0.365203
```

……….

```
 992    3.34000       0.656607E-01   0.490624E-01   0.298281       0.734681       0.633402
 993    3.35000       0.664287E-01   0.496719E-01   0.302427       0.747411       0.653523
```

Describe the model (effects included and their type)

```
 Number of Traits             1
 Number of Effects           34
 Position of Observations    14
 Names of Weight              coef_w for Milk
    coef_w is created from table TDM_test/tab_weight_pet3L and indexed by variable DIMr

 Values less than   -999999.     are considered as missing
 Values larger than  999999.     are considered as missing
 Values equal to     0.00000     are considered as missing
```

```
EFFECTS
  effect HTD is cross-classified
  effect calv_m is cross-classified
  effect calv_a is cross-classified
  effect tar is cross-classified
  effect 51a is a continuous covariable nested within nlac
             whose coefficients come from table TDM_test/tab_spline_DCCpet3L
  effect 52a is a continuous covariable nested within nlac
             whose coefficients come from table TDM_test/tab_spline_DCCpet3L
  ………………………………………………………………………………………………

MODEL
  trait  1  : Milk = HTD + calv_m + calv_a + tar + 51a [ nlac ] + 52a [ nlac ] + 53a [ nlac ] + 54a [ nlac ] +
71a [ mvel_nl ] + 72a [ mvel_nl ] + 73a [ mvel_nl ] + 74a [ mvel_nl ] + 75a [ mvel_nl ] + 76a [ mvel_nl ] + 81a [
tar_nl ] + 82a [ tar_nl ] + 83a [ tar_nl ] + 84a [ tar_nl ] + 85a [ tar_nl ] + 86a [ tar_nl ] + vg1 [ anim ] +
vg2 [ anim ] + vg3 [ anim ] + vg4 [ anim ] + vp1 [ ani_p ] + vp2 [ ani_p ] + vp3 [ ani_p ] + vp4 [ ani_p ] + vh1
[ Hy ] + vh2 [ Hy ] + vh3 [ Hy ] + vh4 [ Hy ] + vh5 [ Hy ] + vh6 [ Hy ]
```

Describe the solving algorithm

```
ALGORITHM
 for solution of mixed model equations: Conjugate gradient iterations storing XpX in core
  Convergence criteria will consider random effects only

 Incomplete Cholesky blocks:
  in block  1:
     effect  1  : HTD
  in block  2:
     effect  2  : calv_m
………………………………………………………………………………………………
```

Specify (Co)variance structure(s) for random effects

```
(CO)VARIANCES
 Residual (co)variance Matrix
 Milk      250.00

 correlated random effects   vg1       vg2       vg3       vg4
 Type of Random Effect:      additive animal with unknown parent groups
    Note: specified covariance matrix is diagonal
 Pedigree File:              TDM_test/pedsim_pet3L.dat

 trait   effect    (CO)VARIANCES
 Milk    vg1         3.3689       0.0000       0.0000       0.0000
 Milk    vg2         0.0000       1.7708       0.0000       0.0000
 Milk    vg3         0.0000       0.0000      0.57778       0.0000
 Milk    vg4         0.0000       0.0000       0.0000      0.18488

 correlated random effects   vp1       vp2       vp3       vp4
 Type of Random Effect:      diagonal
    Note: specified covariance matrix is diagonal

 trait   effect    (CO)VARIANCES
 Milk    vp1         2.3650       0.0000       0.0000       0.0000
 Milk    vp2         0.0000       1.7994       0.0000       0.0000
 Milk    vp3         0.0000       0.0000      0.59212       0.0000
 Milk    vp4         0.0000       0.0000       0.0000      0.49684

 correlated random effects   vh1       vh2       vh3       vh4       vh5       vh6
 Type of Random Effect:      diagonal
    Note: specified covariance matrix is diagonal

 trait   effect    (CO)VARIANCES
 Milk    vh1         4.3291       0.0000       0.0000       0.0000       0.0000       0.0000
 Milk    vh2         0.0000       1.5104       0.0000       0.0000       0.0000       0.0000
 Milk    vh3         0.0000       0.0000       1.3077       0.0000       0.0000       0.0000
 Milk    vh4         0.0000       0.0000       0.0000      0.79933       0.0000       0.0000
 Milk    vh5         0.0000       0.0000       0.0000       0.0000      0.59179       0.0000
 Milk    vh6         0.0000       0.0000       0.0000       0.0000       0.0000      0.43201
```

Describe structure of byprod file

```
BY PRODUCTS
  Storage in file byprod_new of the following variables:
    in column  1 :   ani_p
    in column  2 :   HTD
    in column  3 :   DIMr
    in column  4 :   Milk
    in column  5 :   residual of trait Milk i.e., trait   1
    in column  6 :   standardized residual of trait Milk i.e., trait   1
```

Give important information about the data and pedigree files (number of observations, overall statistics)

```
 Statistics per trait :
  Milk :        mean =   255.8     std =   206.03      (     9736 observations)
                min  =  -584.9     max =   1041.
                mean weight =  0.8972        weight std =  0.27162
                min weight  =  0.2896        max weight =   1.294

  pedigree file = TDM_test/pedsim_pet3L.dat
  pedigree length : 3286  largest animal or unknown parent group number : 3290
   Matrix to store pedigree allocated:  4  x  3290
```

Display first and last lines of pedigree file

```
 First and last 5 pedigree records
 1 862 428 1
 2 1258 427 1
 3 805 748 1
 4 800 749 1
 5 800 750 1
  ....
 3281 3287 3287 3
 3282 3287 3287 3
 3283 3287 3287 3
 3284 3286 3285 1
 3285 3287 3287 3
 3286 3287 3287 3

 Groups of unknown parents from  3287  to  3290
```

Give information about each effect in the model

```
 Number of levels per effect :
  HTD :           2197 (equations        1 to     2197)
  calv_m :          82 (equations     2198 to     2279)
  calv_a :          54 (equations     2280 to     2333)
  tar :             43 (equations     2334 to     2376)
  51a :              1 coefficient(s) *        3   continuous covariate(s), (equations     2377 to     2379) with
                mean =  0.3539     std =  0.47819     min =   0.000     max =   1.000
  52a :              1 coefficient(s) *        3   continuous covariate(s), (equations     2380 to     2382) with
••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
                mean = -0.7979     std =   2.1187     min =  -5.393     max =   5.682
  vh6 :              1 coefficient(s) *      292   continuous covariate(s), (equations    18871 to    19162) with
                mean = -0.9737     std =   1.9243     min =  -8.030     max =   3.745
        => Total number of equations =  19162
 check convergence considering only random effects ( equations     2491 to    19162)
 Storage allocated for the data: (     9736 x  14)
```

Call to the system to know cpu used

```
 read data file again and store
     cpu ==>    ugenvpd 7475222 5693672 A       00:01   00:00:00  0,0  0,0       0   2524 genekit.
 datafile = TDM_test/datsim_pet3L.dat
 end of data storage ( 9736  records stored)
     cpu ==>    ugenvpd 7475222 5693672 A       00:01   00:00:00  0,0  0,0       0   4652 genekit.
```

Preparation (may include reading starting solutions)

```
INITIALIZATION
  initialization of solutions for effect HTD

PREPARATION STEP
  XX has been created
   (maximum number of nonzero elements expected in XX :    500000)
  No elements added to XX before decomposition
  parameters of trait  1  used for the preconditioner
  allocate xlist  500000  = 0  Meg
  allocation OK
     cpu ==>    ugenvpd 7475222 5693672 A       00:02   00:00:00  0,0  0,0       0  16764 genekit.
  create_xx is now finished
  pedigree file read in  0 s,   72743  nonzeroes
  add_random is now finished (or skipped if fixed effect model)
     cpu ==>    ugenvpd 7475222 5693672 A       00:03   00:00:00  0,0  0,0       0  20300 genekit.
  read file in  0 sec
  convergence criterion              :  0.100E-03
  maximum number of iterations       :  500
  start at solution= 0 (True/False)  :  F
  extra terms /line in incomplete Cholesky :  0
```

Computation of the preconditioner when (preconditioned) conjugate gradient is used

```
PRECONDITIONING
  compute preconditioner with alpha =  0.000

  total number of nonzero elements in incomplete Cholesky =  72743
   end of incomplete Cholesky in  0 sec
  preconditioner is now computed (final alpha = 0.000    )
     cpu ==>    ugenvpd 7475222 5693672 A       00:03   00:00:00  0,0  0,0       0  32060 genekit.
  allocate xlist  500000  = 0  Meg
  allocation OK
     cpu ==>    ugenvpd 7475222 5693672 A       00:03   00:00:00  0,0  0,0       0  32060 genekit.
  XX built and stored in  2 sec

  Number of nonzero elements in XX :    464040

     cpu ==>    ugenvpd 7475222 5693672 A       00:06   00:00:03  4,5  0,0       0  34908 genekit.
```

Actual iterations. Note the definition of the convergence criteria

```
ITERATIVE SOLUTION
  right-hand side updated in  0 sec
 Convergence criteria with respect to previous iteration
     average abs(change), standardized norm of residual, max change and equation number
it    1 trait  1 iter   1 ave=  0.101D+02  ||resid||=  0.265D+00  max=  0.230D+02    17699
 solutions stored in file: sol_pet3L
it    2 trait  1 iter   2 ave=  0.763D+02  ||resid||=  0.186D+00  max=  0.366D+02    12354
it    3 trait  1 iter   3 ave=  0.223D+02  ||resid||=  0.819D-01  max=  0.667D+01    17464
it    4 trait  1 iter   4 ave=  0.304D+02  ||resid||=  0.542D-01  max=  0.599D+01    15644
……………………………………………………………………………………………………………………………
it   39 trait  1 iter  39 ave=  0.541D-01  ||resid||=  0.515D-03  max=  0.188D+00     5201
it   40 trait  1 iter  40 ave=  0.244D+00  ||resid||=  0.508D-03  max=  0.184D+00     9036
it   41 trait  1 iter  41 ave=  0.687D-01  ||resid||=  0.541D-03  max=  0.102D+00    17673
```

In the “SOLUTIONS” statement, it was specified that solutions 40 iterations will be stored and compared Note again the definition of convergence criteria 40 iterations apart.

```
 ! ---------------------------------------------------------------------------
 !            Convergence criteria  40  iterations apart
 ! for each effect:  M <==> mean solution;
 !                   S <==> standard deviation of solutions;
 !                   D <==> average change in solutions;
 !                 std <==> standard deviation of change;
 !                   R <==> correlations between solutions;
 !             Min/Max <==> maximum decrease /
 !                             increase in solutions and corresponding level
 ! ---------------------------------------------------------------------------
 solutionfile=sol_pet3L
  41  1 HTD      M=  0.257D+03 S= 0.262D+02 D= -0.661D+01 s= 0.27D+02 R= 0.60321 Min -.197D+03 :    1052 Max
0.120D+03 :    2156
  41  1 calv_m   M=  0.000D+00 S= 0.764D+01 D=  0.000D+00 s= 0.26D+02 R= 0.26464 Min -.733D+02 :      77 Max
0.599D+02 :      49
  41  1 calv_a   M=  0.000D+00 S= 0.317D+01 D=  0.000D+00 s= 0.23D+02 R= 0.33574 Min -.527D+02 :      50 Max
0.655D+02 :       8
  41  1 tar      M=  0.000D+00 S= 0.366D+01 D=  0.000D+00 s= 0.18D+02 R= 0.53804 Min -.645D+02 :      28 Max
0.367D+02 :      34
  41  1 51a      M=  0.270D+00 S= 0.112D+01 D= -0.324D+02 s= 0.11D+02 R= 0.47591 Min -.458D+02 :       3 Max
……………………………………………………………………………………………………………………………
  41  1 vh5      M= -0.703D-03 S= 0.168D+00 D= -0.661D-01 s= 0.11D+01 R= 0.41539 Min -.659D+01 :     289 Max
0.807D+01 :      25
  41  1 vh6      M=  0.615D-03 S= 0.151D+00 D=  0.214D-01 s= 0.74D+00 R= 0.37567 Min -.324D+01 :     225 Max
0.240D+01 :      54
 solutions stored in file: sol_pet3L
it   42 trait  1 iter  42 ave=  0.292D-01  ||resid||=  0.512D-03  max=  0.135D+00    17535
it   43 trait  1 iter  43 ave=  0.287D-01  ||resid||=  0.478D-03  max=  0.111D+00    17535
it   44 trait  1 iter  44 ave=  0.367D-01  ||resid||=  0.479D-03  max=  0.114D+00     5576
```

```
.................................................................................................
it  336 trait  1 iter 336 ave=  0.833D-03  ||resid||=  0.124D-05  max=  0.273D-03    12326
it  337 trait  1 iter 337 ave=  0.132D-03  ||resid||=  0.117D-05  max=  0.185D-03    17679
it  338 trait  1 iter 338 ave=  0.731D-03  ||resid||=  0.112D-05  max=  0.365D-03     8934
it  339 trait  1 iter 339 ave=  0.154D-03  ||resid||=  0.126D-05  max=  0.221D-03    17702
it  340 trait  1 iter 340 ave=  0.133D-03  ||resid||=  0.119D-05  max=  0.261D-03    17523
it  341 trait  1 iter 341 ave=  0.109D-03  ||resid||=  0.111D-05  max=  0.251D-03    17702
it  342 trait  1 iter 342 ave=  0.717D-04  ||resid||=  0.107D-05  max=  0.292D-03     8950
```

The convergence criterion was reached. Final solutions are stored and results are compared with solutions 40 iterations before.

```
  solutions obtained in  3 sec
  ***  System(s) solved in  6 sec
 solutionfile=sol_pet3L
9999  1 HTD      M=  0.257D+03 S= 0.248D+02 D=  0.724D-02 s= 0.55D-01 R= 1.00000 Min -.105D+00 :     421 Max
0.133D+00 :    1044
9999  1 calv_m   M=  0.000D+00 S= 0.803D+01 D=  0.000D+00 s= 0.39D-01 R= 0.99999 Min -.160D+00 :      21 Max
0.811D-01 :      54
9999  1 calv_a   M=  0.000D+00 S= 0.278D+01 D=  0.000D+00 s= 0.19D-01 R= 0.99998 Min -.423D-01 :      21 Max
0.341D-01 :      48
.................................................................................................
9999  1 vg1      M= -0.381D-01 S= 0.431D+01 D= -0.227D-02 s= 0.63D-03 R= 1.00000 Min -.463D-02 :     119 Max
0.140D-02 :    1316
9999  1 vg2      M= -0.182D+01 S= 0.270D+01 D= -0.142D-02 s= 0.67D-03 R= 1.00000 Min -.317D-02 :    3227 Max
0.877D-03 :    2696
9999  1 vg3      M= -0.674D+00 S= 0.674D+00 D=  0.169D-03 s= 0.25D-03 R= 1.00000 Min -.741D-03 :    2696 Max
0.119D-02 :    1316
9999  1 vg4      M= -0.427D+00 S= 0.251D+00 D= -0.147D-02 s= 0.28D-03 R= 1.00000 Min -.277D-02 :    2933 Max -
.872D-03 :    3149
9999  1 vp1      M=  0.409D-03 S= 0.570D+01 D= -0.149D-04 s= 0.52D-03 R= 1.00000 Min -.166D-02 :     180 Max
0.146D-02 :     411
9999  1 vp2      M= -0.360D-03 S= 0.469D+01 D=  0.266D-04 s= 0.44D-03 R= 1.00000 Min -.151D-02 :      76 Max
0.129D-02 :     261
9999  1 vp3      M=  0.448D-03 S= 0.109D+01 D= -0.294D-05 s= 0.23D-03 R= 1.00000 Min -.760D-03 :     405 Max
0.669D-03 :      76
9999  1 vp4      M=  0.273D-03 S= 0.125D+01 D= -0.256D-05 s= 0.25D-03 R= 1.00000 Min -.972D-03 :     405 Max
0.724D-03 :     128
9999  1 vh1      M=  0.122D-03 S= 0.176D+01 D=  0.273D-04 s= 0.80D-03 R= 1.00000 Min -.301D-02 :     103 Max
0.273D-02 :      68
9999  1 vh2      M= -0.146D-03 S= 0.421D+00 D=  0.655D-05 s= 0.22D-03 R= 1.00000 Min -.813D-03 :      65 Max
0.613D-03 :     144
9999  1 vh3      M= -0.779D-04 S= 0.479D+00 D=  0.357D-05 s= 0.31D-03 R= 1.00000 Min -.114D-02 :      50 Max
0.150D-02 :     285
9999  1 vh4      M=  0.204D-03 S= 0.276D+00 D=  0.605D-05 s= 0.18D-03 R= 1.00000 Min -.860D-03 :     264 Max
0.704D-03 :     263
9999  1 vh5      M=  0.372D-03 S= 0.168D+00 D=  0.558D-05 s= 0.10D-03 R= 1.00000 Min -.542D-03 :      64 Max
0.663D-03 :     290
9999  1 vh6      M=  0.107D-03 S= 0.150D+00 D= -0.764D-05 s= 0.96D-04 R= 1.00000 Min -.535D-03 :      63 Max
0.417D-03 :     281
 solutions stored in file: sol_pet3L
```

By-products are computed

```
 by-products of the evaluation stored in file: byprod_new
                                    (= 9737  records)
  total cpu time:  6 sec
```