

D F R E M L

---

— Version 3.0  $\beta$  —

U s e r N o t e s

KARIN MEYER

Last updated September 9, 1998

---

©Karin Meyer 1997, 1998

Permission is given to make and distribute verbatim copies of this document, provided it is preserved complete and unmodified.

---

# Contents

---

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>                       | <b>1</b>  |
| <b>2</b> | <b>Availability</b>                       | <b>3</b>  |
| 2.1      | Conditions of Use . . . . .               | 3         |
| <b>3</b> | <b>Installation</b>                       | <b>4</b>  |
| 3.1      | DOS Systems . . . . .                     | 4         |
| 3.2      | UNIX Systems . . . . .                    | 4         |
| <b>4</b> | <b>Models of Analyses</b>                 | <b>6</b>  |
| 4.1      | Random Effects . . . . .                  | 6         |
| 4.2      | Fixed Effects . . . . .                   | 7         |
| 4.3      | Basic Models . . . . .                    | 8         |
| 4.4      | Repeated Records . . . . .                | 8         |
| 4.5      | Random Regressions . . . . .              | 8         |
| <b>5</b> | <b>The Programs</b>                       | <b>10</b> |
| 5.1      | D F P R E P . . . . .                     | 10        |
| 5.1.1    | Input Files . . . . .                     | 11        |
| 5.1.2    | Interactive Input . . . . .               | 11        |
| 5.1.3    | Tasks performed . . . . .                 | 11        |
| 5.2      | D F U N I . . . . .                       | 12        |
| 5.3      | D X M U X . . . . .                       | 13        |
| 5.4      | D X M R R . . . . .                       | 13        |
| <b>6</b> | <b>Run Options</b>                        | <b>15</b> |
| 6.1      | Set-Up step . . . . .                     | 15        |
| 6.2      | Ordering of Equations . . . . .           | 16        |
| 6.3      | Estimation . . . . .                      | 16        |
| 6.4      | Solutions at Convergence . . . . .        | 17        |
| 6.5      | Find the Best Point So Far . . . . .      | 17        |
| 6.6      | Evaluating Covariance Functions . . . . . | 17        |

|           |  |           |
|-----------|--|-----------|
| <b>7</b>  | <b>Input Files to be Supplied by the User</b>                  | <b>18</b> |
| 7.1       | Pedigree File . . . . .  | 18        |
| 7.2       | Data File . . . . .  | 18        |
| 7.3       | Other Files . . . . .  | 19        |
| 7.3.1     | Arbitrary Covariance Structure for Second Animal Effect . . .  | 19        |
| 7.3.2     | Arbitrary Covariance Structure for an Additional Random Effect | 19        |
| 7.3.3     | User-defined Random Regression . . . . .                       | 19        |
| 7.3.4     | Input of Starting Values from File . . . . .                   | 20        |
| 7.3.5     | Multiple Measurement Error Variances . . . . .                 | 20        |
| <b>8</b>  | <b>Output Files in DFREML</b>                                  | <b>21</b> |
| 8.1       | Results File . . . . .   | 21        |
| 8.2       | Other Output Files . . . . .                                   | 21        |
| 8.2.1     | Monitoring Iterates . . . . .                                  | 21        |
| 8.2.2     | Parameter Estimates . . . . .                                  | 21        |
| 8.2.3     | Breeding value estimates . . . . .                             | 21        |
| 8.2.4     | 'Re-constructed' Covariance Matrices . . . . .                 | 22        |
| <b>9</b>  | <b>Examples</b>  | <b>23</b> |
| 9.1       | Running the Examples . . . . .                                 | 23        |
| 9.1.1     | DOS . . . . .  | 24        |
| 9.1.2     | UNIX . . . . .   | 24        |
| 9.2       | Example 1 . . . . .  | 25        |
| 9.3       | Example 2 . . . . .  | 25        |
| 9.4       | Example 3 . . . . .  | 25        |
| <b>10</b> | <b>References</b>  | <b>26</b> |

---

# 1. Introduction

---

## What is it ?

DFREML is a set of programs to estimate variance and covariance components for continuous traits by Restricted Maximum Likelihood (REML), fitting a linear mixed model.

DFREML has been written for the quantitative genetic analysis of biological data, in particular from animal breeding applications, where we not only want to estimate phenotypic (co)variances but also want to partition them in their additive genetic and other components, and calculate the resulting genetic parameters, e.g. heritabilities and genetic correlations.

In calculating the likelihood it is assumed that the data have a multivariate normal distribution. The model of analysis fitted throughout is the so-called *Animal Model* which includes a random effect representing the additive genetic merit (or breeding value) for each individual (=animal) and trait. Allowing for some additional random effects to be fitted, a total of 10 different models of analysis are accommodated.

## History

DFREML has evolved slowly over the last 10 years, and is likely to change further. The original programs (version 1.0) were written late 1987 to early 1988 at the Institute of Animal Genetics of Edinburgh University in Scotland. Later modifications have been carried out in part while working in Edinburgh and in part at the Animal Genetics and Breeding Unit of the University of New England in Armidale, Australia.

The name, DFREML, is derived from the fact that the REML algorithm used originally to locate the maximum of the likelihood function ( $\log \mathcal{L}$ ) did not require calculating the derivatives of  $\log \mathcal{L}$ ; DFREML standing for **D**erivative - **F**ree **R**estricted **M**aximum **L**ikelihood.

Version 1.0 was first made publicly available in June 1988 and included programs for univariate analyses only (Meyer 1988). Version 2.0, put together in May 1991, implemented multivariate analyses, better ordering of equations in the mixed model array, choice of search strategy, facilities to maximize with respect to a subset of parameters only, approximation of sampling errors and the option to obtain breeding value estimates. In addition, considerable effort was made to improve ease of use by carrying information between programs, fault trapping interactive input and suggesting default answers. Version 2.1 (made available via anonymous ftp in October 1992) then incorporated the sparse matrix routines of George and Liu (1991) to factor the mixed model matrix.

An “ $\alpha$ ” version of DFREML 3.0 has been put together in November 1997, for inclusion on the CD of the 6<sup>th</sup> World Congress on Genetics Applied to Livestock Production. Updates fixing a number of little inconsistencies and problems with different platforms as well as expanding the options available have been made available for downloading on a continuous basis since then.

## What is new in version 3.0 ?

A major tidy-up took place in compiling version 3.0. This included elimination of little used run-options and of one of the multivariate programs.

The major new features of version 3.0 are :

1. Programs are written in FORTRAN 90.
  - Using dynamic array allocation, programs minimize the amount of memory required.
  - Frequent recompilation to adapt programs for specific data sets and analyses is unnecessary.
2. There are a platform specific versions :
  - a UNIX version consisting of FORTRAN 90 source code and accompanying “Makefiles”
  - A PC (DOS) version consisting of executable files, created for a pentium processor using Lahey’s LF90 compiler.
3. There is a choice of procedure to maximize  $\log \mathcal{L}$ 
  - Derivative-free searches of former versions are still available.
  - The so-called “average information” (AI-REML) algorithm (Johnston and Thompson 1995) has been implemented.
4. Approximate sampling errors of estimates of covariance components and genetic parameters are derived from the average information matrix.
5. Reparameterization (optional) to the Cholesky decomposition of covariance matrices to be estimated is available.
6. A program for the estimation of covariance functions for longitudinal data (“repeated” records) fitting a random regression model is included.
7. Estimation of reduced rank covariance matrices is supported.

---

## 2. Availability

---

An "α" version of DFREML 3.0 has been made available on the CD produced for the 6<sup>th</sup> World Congress on Genetics Applied to Livestock Production, Armidale, January 12-16, 1998. Minor modifications have been made since then (some bug fixes). The updated version is available through the DFREML home page, <http://agbu.une.edu.au/~kmeyer/dfreml.html>. Alternatively, all material can be obtained via anonymous ftp from [metz.une.edu.au](ftp://metz.une.edu.au) (129.180.1.4), directory /pub/agbu/DFREML (see README file for details).

The complete package consists of source code, DOS executables, test data, worked examples and this manual.

### 2.1 Conditions of Use

DFREML is distributed free of charge to the scientific community under the conditions that it remains my copyright, that it is not modified other than to adapt it to the local computing environment or for personal research, that it is used for scientific purposes only, and that its use is credited in any publications.

DFREML can be redistributed freely within the scientific community, provided this is done using copies of the original (i.e. complete and unmodified) files.

While every effort has been made to ensure that DFREML does what it claims to do, there is absolutely no guarantee for its correctness. You are using DFREML entirely at your own risk, and there is no user-support service.

I do invite constructive criticism and genuine bug reports for this "β" version though to : [kmeyer@didgeridoo.une.edu.au](mailto:kmeyer@didgeridoo.une.edu.au)

---

## 3. Installation

---

### 3.1 DOS Systems

The DOS version of DFREML consists of executable files. Installation is easy, though not totally automatic.

1. Download the latest version as `dfreml.v30.dos.exe` from the web/ftp site.
2. Create the target directory for DFREML default `C:DFREML`.
3. Extract `dfprep.exe`, `dfuni.exe`, `dxmux.exe` and `dxmrr.exe` by double-clicking on the icon for `dfreml.v30.dos.exe` (Windows95).
4. Add directory DFREML to the PATH statement in your `autoexec.bat` file.

### 3.2 UNIX Systems

Installation in a UNIX environment involves compiling the programs in DFREML - a FORTRAN 90 compiler is required !

1. Download the archive `dfreml.v30.tar.gz` which contains the source code and Makefiles for the latest version of DFREML from the web/ftp site.
2. Uncompress the archive : `gunzip dfreml.v30.tar.gz`
3. Extract the files and directories `tar xvf dfreml.v30.tar`

This creates directory DFREML with subdirectories LIB, MRR, MUX, PRE and UNI.

4. Build the library of subroutines `libkm90.a`
  - (a) Change to directory LIB : `cd DFREML/LIB`
  - (b) Compile all subroutines : `f90 -c *.f`
  - (c) Assemble library : `ar rv libkm90.a *.o`
  - (d) Place library into chosen directory, denoted as \$LIBDIR in the following : `mv libkm90.a $LIBDIR`
5. For each of the four programs/subdirectories (PRE, UNI, MUX, MRR) :
  - (a) Change into the subdirectory
  - (b) Edit the marked section of Makefile, specifically setting \$BINDIR and \$LIBDIR according to the local environment (also check compiler name and options).
  - (c) Compile modules file (for some systems this occurs automatically when invoking make) : `f90 -c modules.f90`
  - (d) Build program by simply typing `make`



This will place the appropriate program (dfprep, dfuni, dxmux or dxmrr) in \$BINDIR

6. Add \$BINDIR to your path (login file, etc.).

---

## 4. Models of Analyses

---

This chapter gives an outline of the models of analyses which are accommodated by DFREML.

The main purpose of DFREML is the *estimation of variance and covariance components* and of the corresponding genetic parameters by Restricted Maximum Likelihood. The principal features of the underlying theory and computational steps involved are described in chapter (yet to be written). DFREML is geared towards the analysis of data from animal breeding, but should be equally useful in other, genetically orientated branches of applied biology.

Traits analyzed are assumed to be continuous and to have a multivariate normal distribution. The model of analysis is linear and includes a random effect for the additive genetic value for the ‘experimental unit’ on which the measurement has been taken and each trait.

In animal breeding, these units are generally animals and the model is hence referred to as an ‘Animal Model’. The animal model allows for both animals in the data and parents without records to be included in the analysis so that all known relationships between animals can be taken into account. By including other effects, fixed and random, the basic animal model can be expanded to accommodate a wide range of problems.

### 4.1 Random Effects

As emphasized above, the basic model of analysis invoked by DFREML is the animal model which includes a random effect for the additive genetic merit of each animal and trait in the analysis. In addition, DFREML allows other random effects to be fitted.

Let the vector of additive genetic effects be denoted by  $\mathbf{a}$  and let  $\Sigma_A$  be the matrix of additive genetic covariances for the  $q$  traits analyzed. With  $\mathbf{A}$  equal to the numerator relationship matrix between animals, the covariance matrix of  $\mathbf{a}$  is given by  $\Sigma_A * \mathbf{A}$  (for levels of  $\mathbf{a}$  ordered according to animals within traits).

Often, groups of animals are subject to some common environmental effect. Let this be denoted by  $\mathbf{c}$  and let  $\Sigma_C$  be the  $q \times q$  matrix of covariance components for this effect. A typical example is a litter effect in pigs or mice. This effect is usually assumed to be uncorrelated to all other random factors in the model and within traits to be independently and identically distributed. This implies that its covariance matrix is diagonal (univariate analyses) or block-diagonal (multivariate analyses with levels of  $\mathbf{c}$  ordered according to traits within effects). DFREML can take such effects into account as so-called ‘*uncorrelated, additional*’ random effect.

For univariate analyses, DFREML allows several of these effects to be fitted. In addition, the first effect ( $\mathbf{c}_1$ ) is allowed to have an arbitrary covariance matrix  $\Sigma_{C1}\mathbf{B}$ . If this option is chosen, however, DFREML requires the user to supply a file with the non-zero elements of the inverse of this matrix,  $\mathbf{B}^{-1}$ . For multivariate analyses, a

Table 4.1: Models of Analysis in DFREML

| Model No. | Effects Fitted | Variiances Estimated                                  | Max. No. of Parameters <sup>a</sup> | Assumptions  |
|-----------|----------------|---|-------------------------------------|--|
| 1         | <b>a</b>       | $\Sigma_A, \Sigma_E$                                  | $q(q+1)$                            |  |
| 2         | <b>a, c</b>    | $\Sigma_A, \Sigma_C, \Sigma_E$                        | $3q(q+1)/2$                         | $Var(\mathbf{c}) = \Sigma_C * \mathbf{I}$  |
| 3         | <b>a, m</b>    | $\Sigma_A, \Sigma_M, \Sigma_E$                        | $3q(q+1)/2$                         | $Var(\mathbf{m}) = \Sigma_M * \mathbf{A}$  |
| 4         | <b>a, m</b>    | $\Sigma_A, \Sigma_M, \Sigma_{AM}, \Sigma_E$           | $q(5q+3)/2$                         | $Var(\mathbf{m}) = \Sigma_M * \mathbf{A}$<br>$Cov(\mathbf{a}, \mathbf{m}') = \Sigma_{AM} * \mathbf{A}$ |
| 5         | <b>a, m</b>    | $\Sigma_A, \Sigma_M, \Sigma_E$                        | $3q(q+1)/2$                         | $Var(\mathbf{m}) = \Sigma_M * \mathbf{I}$  |
| 6         | <b>a, m</b>    | $\Sigma_A, \Sigma_M, \Sigma_E$                        | $3q(q+1)/2$                         | $Var(\mathbf{m}) = \Sigma_M * \mathbf{D}$  |
| 7         | <b>a, m, c</b> | $\Sigma_A, \Sigma_M, \Sigma_C, \Sigma_E$              | $2q(q+1)$                           | as for models 2 & 3  |
| 8         | <b>a, m, c</b> | $\Sigma_A, \Sigma_M, \Sigma_{AM}, \Sigma_C, \Sigma_E$ | $q(3q+2)$                           | as for models 2 & 4  |
| 9         | <b>a, m, c</b> | $\Sigma_A, \Sigma_M, \Sigma_C, \Sigma_E$              | $2q(q+1)$                           | as for models 2 & 5  |
| 10        | <b>a, m, c</b> | $\Sigma_A, \Sigma_M, \Sigma_C, \Sigma_E$              | $2q(q+1)$                           | as for models 2 & 6  |

<sup>a</sup>to be estimated for  $q$  traits in the analysis

maximum of two such effects are accommodated. Both must have covariance matrix proportional to the identity matrix. If a second, additional random effect is fitted, both must be fitted for all traits.

Furthermore, a ‘*second random effect per animal*’ (denoted by  $\mathbf{m}$  with corresponding matrix of covariances  $\Sigma_M$ ) can be fitted. The number of levels for this effect is assumed to be equal to the number of animals. Examples are maternal or paternal genetic effects, permanent environmental effects due to the animal for traits with repeated records or a dominance effect. A choice of three options for the assumed covariance structure of  $\mathbf{m}$  is provided to accommodate a variety of such effects.

Under option “1”, suitable for a parental genetic effect, it is assumed that  $\mathbf{m}$  has the same covariance structure as the vector of additive genetic effects  $\mathbf{a}$ . Option “2” implies, as outlined for  $\mathbf{c}$  above, that  $\mathbf{m}$  is uncorrelated to all other random effects in the model with covariance matrix proportional to the identity matrix. Option “3” finally assumes, as for option “2”, that  $\mathbf{m}$  is independent of  $\mathbf{a}$  and  $\mathbf{c}$  but allows for covariances between levels of  $\mathbf{m}$ . Let  $\mathbf{D}$  be the arbitrary covariance matrix for the levels of  $\mathbf{m}$  for each trait. DFREML then requires the user to supply a file with the non-zero elements of  $\mathbf{D}^{-1}$ .

## 4.2 Fixed Effects

To account for systematic effects, DFREML allows several fixed effects and covariables to be fitted. Fixed effects are assumed to be cross-classified. To account for a hierarchical structure, the user needs to specify the additional dependencies among fixed effects level explicitly. Currently there is no ‘automatic’ coding facility for interactions between fixed effects, i.e. to include an interaction the user needs to create an additional fixed effect with the appropriate coding and to specify the dependencies arising as for nested fixed effects.

Except for univariate analyses, covariables are fitted across all effects. Linear and higher order regression coefficients can be fitted. Only “DFUNI” accommodates

regressions within fixed effects. If this is chosen, the fixed effect(s) within which regressions are fitted must explicitly be included in the model of analysis.

### 4.3 Basic Models

Various combinations of the effects fitted and covariance options chosen then yield a total of 10 different models accommodated by DFREML. These are summarized in Table 4.1, where the ‘maximum number of parameters’ refers to analyses without ‘missing’ or non-estimable covariances, for instance due to different traits being measured on different subsets of animals.

### 4.4 Repeated Records

If there are repeated records per trait, the permanent environment due to the animal can be taken into account by fitting an appropriate random effect. This can be done, as outlined above, by fitting a second random effect for each animal (Model 5 or 9). Alternatively, it could be included as additional, uncorrelated random effect (Model 2 or 7–10), in particular if the second animal effect is already used to represent some other effect.

Furthermore, repeated records per animal can be modeled by allowing for a covariance among residuals. This yields a so-called ‘equivalent model’. In terms of estimation though, it is somewhat different in such that the non-negativity constraint imposed on REML estimates of variance components is removed. DFREML provides the option to allow for repeated records in this way. This can be advantageous not only in reducing the total number of effects in the model of analysis but also in improving convergence of the iterative estimation procedure.

The equivalent repeatability model has been found especially useful for bivariate analyses involving one trait with single records only and one trait with repeated records. Unless these traits were measured on distinct subsets of animals (so that all environmental covariances were zero), an explicit second random effect would have to be fitted for both traits. Consequently, it would be attempted to separate the permanent and temporary environmental variance for both traits. For the trait with single records, these have a high negative sampling correlations and problems in locating the maximum of the likelihood have been encountered, in particular for data sets with a relatively small proportion of animals with records for both traits. Under the equivalent model, however, the likelihood can be maximized with respect to the total environmental variance for the single record trait, thus reducing the number of parameters to be estimated by one with a corresponding number of likelihood evaluations required.

### 4.5 Random Regressions

The description of models so far has assumed a “finite-dimensional” model, where records on the same animal are either different traits or repeated measurements of the same trait, and we fit a separate random effect for each trait and level of  $\mathbf{a}$ ,  $\mathbf{c}$  and  $\mathbf{m}$ . Recently, however, an “infinite-dimensional” model has been suggested (Kirkpatrick *et al.*1990). This is suited to longitudinal (or equivalent) data where we have ‘repeated’ records for a trait on an individual along some continuous scale (‘meta-meter’) such as time, and want to allow for the fact that the trait in question

might be continually and gradually changing. A typical example is the growth of an animal, with weight measured at different ages.

This can be modeled by fitting a set of random regression coefficients on functions of age (or equivalent) for each animal, instead of a single random effect (univariate repeatability model) or a random effect for each age (finite-dimensional, multivariate model). Such random regression coefficients can be fitted for each source of variation. Modeling both additive genetic and permanent environmental effects in this way, we can estimate temporary environmental or measurement error variances without the need to have multiple records for each age.

Estimating the covariances among the random regression coefficients then yields estimates of *Covariance Functions*, i.e. a continuous description of the covariance structure of the trait for the range of ages (or equivalent) spanned by the data. Similarly, estimates of the additive genetic random regression coefficients for an animal provide a description of its complete growth curve (more generally trajectory).

DFREML includes a program for such analyses. Currently, only two sets of random regression coefficients – pertaining to additive genetic effects and permanent environmental effects due to the animal – can be fitted, but multiple measurement error variances are accommodated.

---

## 5. The Programs

---

Version 3.0 of DFREML consists of a total of 4 programs, see Table 5.1. These are a general ‘preparation’ program, “DFPREP”, and 3 specific estimation programs, namely “DFUNI” for univariate analyses, “DXMUX” for multivariate analyses, and “DXMRR” to estimate covariance functions for “repeated” records.

Each of the 3 estimation programs offers a number of *run options* (see section 6 below), allowing analyses to be broken into distinct steps and providing facilities for additional calculations.

Most information needed by the programs is acquired as interactive input, i.e. apart from the data and pedigree files (described below, see sections 7.1 and 7.2), the user generally does not have to set up any files (exceptions apply for ‘non-standard’ analyses; see section 7.3), though for convenience “DXMUX” and “DXMRR” accept information on starting values from either file or standard input. As far as possible, interactive input has been made easy by fault-trapping input (so that, for instance, a character typed in a numerical field will not cause programs to crash) and by suggesting default answers.

DFREML creates a number of *“internal” files*, mainly unformatted, to store and transfer information between individual steps. These generally have standard names DFmn#DAT where mn is the logical unit number. They are described individually in chapter ?? (yet to be written); as a rule, however, the user does not need to be concerned with them, provided there is sufficient disk space.

Table 5.1: Programs in DFREML v3.0

| Program  | Purpose                 | Details  |
|----------|-------------------------|--|
| “DFPREP” | Preparation step        | Recode fixed and random effects in running order and set up the NRM between animals                            |
| “DFUNI”  | Univariate analyses     | Derivative-free algorithm only   |
| “DXMUX”  | Multivariate analyses   | AI-REML and DF-REML algorithm; different models for different traits; accommodate non-estimable covariances    |
| “DXMRR”  | Covariance function an. | AI-REML and DF-REML algorithm; “repeated” records for single trait; estimate reduced rank covariance functions |

### 5.1 D F P R E P

This program precedes each of the three estimation programs. It performs a series of ‘data preparation’ steps, in particular recoding of animals and other effects levels in running order and setting up of the numerator relationship between animals. In addition, there are ‘special effects’ options, such as the facility to insert the code for an additional random effect due to animals or to ‘prune’ pedigrees (i.e. eliminate parents not in the data with single offspring).

### 5.1.1 Input Files

As described above, “DFPREP” generally requires a *pedigree file* (see section 7.1) and a *data file* (see section 7.2) as input. Only for univariate analyses with single records per animal can these two be the same, and only for analyses with “DXMRR” estimating phenotypic (rather than genetic and permanent environmental) covariance functions can the pedigree file be missing.

“DFPREP” can read these as binary files (unformatted READ), column separated formatted files (list-directed READ), or as formatted files in a user-specified format. All codes are expected to be full word `INTEGER(4)` variables, and all values for traits and covariables are read as `REAL(8)` variables.

### 5.1.2 Interactive Input

“DFPREP” requires the following information :

- The name of the pedigree file and its format.
- An upper limit for the numerical value of animal identities.
- An option specifying the estimation program to be used.
- The name of the data file and its format.
- The number of traits in the analysis and/or the number of right hand sides per record.
- The number of covariables to be fitted for each trait.
- The number of fixed effects to be fitted for each trait.
- The number of additional random effects to be fitted for each trait.
- Whether a second random effect for each animal is fitted, whether it is to be added to each record, and whether this is the animal, sire or dam code.
- An option specifying whether any ‘special’ action is required and any details related to it.
- The maximum number of animals in the analysis (including parents in the analysis); this does not need to be very accurate, though too big a number might cause “DFPREP” to fail if arrays cannot be allocated.
- An option specifying whether the setting up the inverse of the numerator relationship matrix (NRM) can be skipped - ONLY if a file `DF44#DAT` exists and the numbers of animals are consistent.

### 5.1.3 Tasks performed

After interactively acquiring information on record layout, input files and type of analysis, the program performs one loop through the pedigree file to compile a list of all animal identities. This is sorted in ascending order. In a second pedigree loop, parental identities for all animals recoded to running order are established and numbers of offspring per animal are counted. The program performs several pedigree checks, stopping if :

1. An animal code higher than the specified upper limit is found.

2. The animal identity is lower than that of either parent.
3. Sire and dam identities are the same
4. A parent ‘changes sex’, i.e. an animal which is first encountered as a sire (dam) subsequently occurs as a dam (sire).

Going through the data file, it is then determined which animals have records. Subsequently, it is attempted to ‘prune’ the pedigree : All animals without records (= parents) and single links to other animals, i.e. one offspring only and unknown parents, do not add any information to the analysis and can be treated as unknown parents. Conceptually, this is equivalent to absorbing the equations for these animals in the mixed model matrix. As many pruning loops as necessary are performed, until no more animals are fulfilling the above criteria are found. If a parental genetic effect is fitted, however, parents of the appropriate sex should not be pruned.

If animals have been pruned, animal and parental identities are recoded to the new running order before setting up the inverse of the NRM (non-zero elements only) using the algorithm given by Quaas (1976)<sup>1</sup>. This yields inbreeding coefficients and the log determinant of the matrix as by-products. Before setting up the NRM inverse, “DFPREP” checks whether a file DF44#DAT, i.e. the DFREML file which stores the coefficients of this matrix, exists in the current directory. If it does and if the number of animals in the analysis read from DF44#DAT agrees with the current number of animals, the user is given the option to skip this time-consuming step.

In a second loop through the data file, codes for the levels of all fixed effects and additional random effect are collected and sorted in ascending order. Rereading the data, a file is written out with animal identities and codes for all effects fitted recoded to running order. If specified, a code for a second random effect per animal is inserted. At the same time, numbers of records and total for the first right hand side per record are accumulated.

Finally, various summary statistics are calculated and information on characteristics of the data and pedigree structure, average inbreeding coefficients if requested, phenotypic means and standard deviations for traits and covariables, and codes, numbers and means for fixed effects levels is written to the results file.

## 5.2 D F U N I

This program performs simple, *univariate* analyses using a derivative-free algorithm. In essence, “DFUNI” of version 3.0 is a cut-down variant of its predecessors in earlier versions. Univariate analyses using the “average information” algorithm can be carried out using “DXMUX” and specifying the number of traits to be 1.

“DFUNI” has been maintained for consistency with older versions of DFREML and as it provides a robust tool with low memory requirements for exploratory analyses. To aid identification of the correct model, “DFUNI” allows additional random effects on the record and specified as such in “DFPREP” to be ignored simply by giving the corresponding number(s) of levels as zero. This allows different models of analysis (sub-models) to be fitted without changing the data file and running “DFPREP” again. In addition, “DFUNI” allows for several uncorrelated, additional random effects while “DXMUX” only caters for a maximum of two such effects. Last but

<sup>1</sup>to be replaced by faster procedure in due course



not least, “DFUNI” currently is the only program in DFREML which allows for covariables to be nested within fixed effects<sup>2</sup>.

As described in detail in Meyer (1989b), the error variance is estimated directly from the residual sum of squares and the likelihood is maximized with respect to the remaining (co)variance components expressed as proportion of the total, phenotypic variance. For a ‘simple’ animal model, this implies a one-dimensional search which is performed using a quadratic approximation of  $\log \mathcal{L}$  as suggested by Graser *et al.* (1987).

The program allows for multiple right hand sides and carries these through to the estimation step(s), so that analyses not only require the set-ups step to be carried only once, but also so that each factorization of the mixed model matrix yields  $\log \mathcal{L}$  for all right hand sides. These are stored and recovered subsequently from file DF59#DAT to find starting values for analyses of trait 2 onwards.

### 5.3 D X M U X

This program performs *multivariate* analyses. It allows different models to be fitted for individual traits, differing in both fixed and random effects. This implies that the interactive information specifying the model of analysis has to be given for each trait separately. “DxMUX” allows the number of traits to be analysed to be equal to 1, i.e. allows univariate analyses to be performed employing an “average information” algorithms<sup>1</sup>.

The program checks and accounts for (co)variance components which cannot be estimated because the data does not supply the necessary information. For instance, when traits are measured on different animals, the error covariance between traits is zero. In addition, it allows covariances to be set to zero on input, recognizes that these are ‘fixed’ and reduces the dimension of search for the maximum of the likelihood accordingly.

“DxMUX” allows the choice between an AI-REML algorithm or a derivative-free search (utilizing either Powell’s (1965) method or Nelder and Mead’s (1965) Simplex procedure). To constrain estimates to the parameter space or speed up convergence, reparameterisation to Cholesky decompositions of the covariance matrices to be estimated (including log transformation of diagonals) can be selected. An option to force estimated covariance matrices to be of reduced rank is planned.

### 5.4 D X M R R

This program performs *covariance function* (CF) analyses for “repeated” records (longitudinal data).

CFs can be estimated by fitting a mixed ‘linear’ model with regression coefficients on some function of time as *random* effects. A set of random regression (RR) coefficients is fitted for each source of variation to be modeled. The coefficients of the CFs are estimated as the covariances among the RR coefficients, using a ‘standard’ multivariate REML algorithm. Regressing on orthogonal (Legendre) polynomials of time yields the CF model described by Kirkpatrick *et al.* (1990). This is the default

<sup>2</sup>option to be implemented for “DxMUX” and “DxMRR” in due course

<sup>1</sup>Output from “DFPREP” specifying a univariate analysis is now suitable for input to “DxMUX”

in “DxMRR”, but other, user-defined functions can be fitted. As for “DxMUX”, both an AI-REML and derivative-free algorithms are available in “DxMRR”.

At present, “DxMRR” does not accommodate additional random effects other than permanent environmental effects due to animals (in addition to animals’ direct, additive genetic effects). Extensions to models incorporating other random effects (e.g. maternal effects) are planned.

It is assumed that the order of polynomial fit is less than the number of time points in the data. This allows temporary and permanent environmental variances to be estimated. It implies that at least two sets of RR coefficients, modeling additive genetic and permanent environmental effects due to the animal, are fitted and corresponding CF are estimated. A “phenotypic” covariance function analysis can be carried out though by fitting a single animal effect (rather than separate genetic and permanent environmental effect) for each animal, while ignoring any relationships between animals. Temporary environmental or measurement error variances are assumed to be independently distributed; “DxMRR” allows for single or multiple error variances to be estimated.

“DxMRR” accommodates different orders of fit for different, uncorrelated CF, thus facilitating the minimum degree of polynomial fit for each CF to be selected. As for “DxMUX”, estimation can be carried out on the original scale or by reparameterising to the Cholesky decompositions of the covariance matrices of RR coefficients. With the latter, an option for estimation of reduced rank covariance matrices (of RR) and thus CFs is available. For a CF with order of fit  $k$  and rank  $m$  this reduces the number of parameters to be estimated from  $k(k + 1)/2$  to  $km - m(m - 1)/2$ . It allows parsimonious modeling when a high degree ( $k$ ) polynomial is needed to describe the shape of trajectory adequately, but only the first  $m$  eigenvalues and eigenfunctions of the CF are important.

Output from “DxMRR” includes the estimated covariances among RR coefficients, the resulting CF and their eigenvalues and eigenfunctions. Optionally, covariance and correlation matrices for the ages in the data are calculated from the estimated CF. Run options are provided to generate a grid of predicted covariances for selected time points, and to obtain backsolutions for the RR coefficients fitted. For the genetic CF, these are predicted breeding values for the shape parameters of the trajectory, and predicted breeding values for selected time points can be calculated from them.

---

## 6. Run Options

---

“DFUNI”, “DxMUX” and “DxMRR” have up to six run options; see Table 6.1. There are two pre-estimation steps (run options “-1” and “-2”), one options for estimation *per se* (option “0”), two options (“5” and “10”) for post-estimation calculations, and one auxiliary ‘print’ option (“8”).

Table 6.1: Summary of run options in DFREML v3.0

| Option | Purpose         | Comments  |
|--------|-----------------|---|
| -1     | Set-Up Step     | Acquire information on model of analysis, check codes & set up files on UNIT “51” and UNIT “52”.  |
| -2     | Order Equations | Re-order equations in the MMM to reduce fill-in arising in factoring the coefficient matrix.      |
| 0      | Estimation      | Read in starting values for parameters, choose search strategy and carry out estimation.          |
| 5      | Solutions       | Obtain solutions for fixed or random effects fitted for current (co)variance component estimates. |
| 8      | Best point      | Pick out point with highest likelihood found so far and print out abbreviated results.            |
| 10     | Grid            | Evaluate grid of covariances for selected covariance function and ages.                           |

### 6.1 Set-Up step

This is the mandatory *first* step for each analysis (after the ‘data preparation’ step via “DFPREP”). It is selected by specifying a run option of “-1”.

#### Tasks performed

Programs first acquire all information needed to determine the model of analysis interactively. For this, output from “DFPREP” specifying numbers of traits and covariables and numbers of fixed and random effects level found, is picked up and this information is used to suggest ‘default’ answers to a number of questions asked.

After the model of analysis has been determined, the data is read checking all effects codes against the number of levels specified for them. All right hand sides (traits) and covariables are deviated from their phenotypic means. The codes for all fixed and random effects fitted are recoded to row numbers in the MMM. All records for an animal are collected, and the animal is assigned a code describing the combination of traits measured and number of records for each trait. If higher order regression coefficients for covariables are fitted, the appropriate exponentials are calculated. A new ‘data file’ with one record per animal is written.

At the end of the set-up step, data is re-read and the least-squares equations for fixed effects and covariables are constructed. These are solved by Gaussian Elimination

and backsolving, identifying at the same time any additional rank deficiencies in the coefficient matrix not taken into account so far, for instance due to confounding of fixed effects. If applicable, the appropriate equation numbers are added to the list of rows and columns to be ‘zeroed’ out, thus identifying a full rank submatrix of the coefficient matrix.

## 6.2 Ordering of Equations

This is the *second* pre-estimation step for each analysis. It is invoked by specifying run option “-2”.

### Tasks performed

When setting up the mixed model array, equations are numbered consecutively in the following ‘standard’ way :

1. One row for each right hand side
2. One row for each (fixed) regression coefficient fitted
3. One row for each fixed effect level; for multivariate analyses levels are ordered within fixed effects and these in turn are ordered within traits
4. (a) For “DFUNI” and “DXMUX”, one row for each trait and level of the additional random effect(s).
- (b) For “DXMRR”, one row for each random regression coefficient and animal, representing the permanent environmental effect of the animal.
5. (a) For “DFUNI” and “DXMUX” : One row for each animal effect fitted and trait, ordering according to traits within effects. If a second animal effect is included in the model of analysis, the corresponding row(s) for each animal immediately follow the one(s) representing the animal’s additive genetic value(s).
- (b) For “DXMRR” : One row for each additive genetic random regression and animal.

Animals are processed in running order (as established by “DFPREP”), i.e. from lowest to highest identity.

In this step, equations are re-ordered using George and Liu’s (1981) subroutines for a minimum degree algorithm which establishes a set of permutations of the original matrix. This is followed by a symbolic factorization step, establishing the data structure of the factored matrix and setting up a compressed storage scheme for the elements of the sparse matrix. See Chapter 5 in George and Liu’s (1981) book for further details and the relevant FORTRAN source code.

When the new order of equations has been established, the data preparation part of the Set-Up step is repeated (see above, section 6.1), recoding effects to their new row numbers, and a vector containing the new order is written to the file containing the general information on the analysis to be performed (UNIT “51”).

## 6.3 Estimation

Estimation is carried out by specifying a run option of “0”.

Programs first read starting values for the parameters to be estimated and an operational zero. Interactive input is then required to specify the procedure to maximize  $\log \mathcal{L}$ , the parameterization to be used, convergence criteria, the number of iterates allowed and any parameters specific to the algorithm, such as step size(s).

During estimation, all points on the likelihood surface evaluated are saved to a file. For any subsequent runs under option “0”, this file is scanned and, if appropriate, the vector of starting values specified is replaced with the best point found. This facilitates re-starts as well as recovery of (most) work carried out by a crashed run.

At convergence or the end of iterations, estimates of (co)variance components and the resulting genetic parameters are written to the results file. If an AI-REML algorithm has been used to maximize  $\log \mathcal{L}$ , approximate sampling errors are obtained automatically from the inverse of the average information matrix. Otherwise the user is given the option to calculate this matrix which involves computational requirements

## 6.4 Solutions at Convergence

This step is chosen by specifying run option “5”.

If found, the program reads in solutions for all effects fitted written at the end of a run under option “0”. If this does not exist, it attempts to calculate them for the best, previously evaluated point found. This involves one likelihood evaluation.

Solutions for fixed effects and covariables and, if selected, all random effects in the model of analysis are then written to the results file `DF66#DAT`. This can result in a large output file !

## 6.5 Find the Best Point So Far

This option allows the best point among all those evaluated so far for the current analysis to be picked out, giving results as if this were the maximum of the likelihood. This is useful to monitor the progress of an analysis running in background mode, for crashed runs to see where one ‘stands’, or for multivariate analyses converging to the bounds of the parameter space to establish whether fixing any correlations to unity might be advantageous.

It is chosen by specifying run option “8”, and a program can be run with option “8” simultaneously to running with option “0”.

## 6.6 Evaluating Covariance Functions

This option, available for “DXMRR” only, allows estimated covariance function(s) to be evaluated for selected points (ages, or equivalent). It is chosen by specifying run option “10”.

---

## 7. Input Files to be Supplied by the User

---

### 7.1 Pedigree File

The pedigree file must contain one record for each animal occurring in the data file. Additional pedigree information, i.e. for parents without records, is optional and can be included by adding respective records. It can be formatted or unformatted, and no specific order of records is required.

Each record must consist of at least 3 `INTEGER` variables :

1. A numerical code identifying the animal;
2. A numerical code identifying the animal's sire; and
3. A numerical code identifying the animal's dam.

Unknown parental identities are to be coded as "0". Animal codes must be of larger numerical values than those of both their parents. For some analyses, pedigree records may need to contain one or two additional integer variables :

4. A code assigning the animal to some group, e.g. year or generation. This is required, if the average inbreeding coefficient (and potentially the average breeding value estimate later on) within levels of this group is to be calculated. The code given must be in running order, i.e. range from 1 to the 'number of level' as specified in response to the respective program prompt.
5. A code describing whether the animal is to be treated as fixed or random for the analysis. This is only required when selected animals are to be treated as fixed (special option "3"). The code should be zero for these animals, and some value greater than zero for 'normal' animals, i.e. to be treated as random.

### 7.2 Data File

The layout of the data file required varies with the model to be fitted and the type of analysis to be carried out. Each record should contain the following variables :

1. An integer code for the trait, in running order from 1 to the number of traits analysed. This is NOT required only for "DFUNI"; it must be set to 1 throughout for "DXMRR".
2. An integer code identifying the animal; this (and the parental codes) must be identical to those given in the pedigree file; mandatory.
3. An `INTEGER` code identifying the animal's sire; mandatory.
4. An `INTEGER` code identifying the animal's dam; mandatory.
5. An `INTEGER` code for each fixed effect to be fitted; optional.
6. An `INTEGER` code for each additional, uncorrelated random effect to be fitted; optional for "DFUNI" and "DXMUX", mandatory for "DXMRR".

7. An `INTEGER` code for a second random effect to be fitted for each animal; optional
8. A `REAL(8)` variable for each covariable to be fitted; optional
9. A `REAL(8)` variable for right hand side (Note the distinction to traits - different traits must be given on different records).
10. An `INTEGER` code for the “meta-meter” in estimating covariance functions, i.e. the continuous scale (such as time, age, distance) along which repeated measurements are taken; required only for “`DXMRR`”.

Records are expected to be ordered according to animals and, for multivariate analyses, traits within animals.

## 7.3 Other Files

Several other files may need to be supplied by the user for some analyses. In addition, some interactive input may optionally be read from a file.

### 7.3.1 Arbitrary Covariance Structure for Second Animal Effect

For analyses fitting a ‘second animal effect’ with covariance matrix not proportional to the numerator relationship matrix or the identity matrix (Models 6 or 10 only), the user needs to supply the inverse of this arbitrary matrix. An example would be a dominance effect, for which the inverse of the dominance relationship matrix is required.

Programs expect to read this information from an unformatted file with standard name `DF45#DAT`. It must contain one record for each non-zero element in the lower triangle of the inverse matrix, each record consisting of 3 variables :

1. An `INTEGER` code giving the column number in the matrix (coded from 1 to number of animals in the analysis);
2. An `INTEGER` code giving the row number (greater or equal than the column number) in the matrix (coded from 1 to number of animals in the analysis); and
3. A `REAL(8)` variable giving the coefficient in the matrix.

### 7.3.2 Arbitrary Covariance Structure for an Additional Random Effect

Similarly, an arbitrary covariance matrix (rather than the identity matrix) for the first additional, uncorrelated random effect fitted can be accommodated in “`DFUNI`”. As above (section 7.3.1), the user needs to supply this inverse (non-zero elements, lower triangle only). It is assumed to be held in the unformatted file `DF47#DAT` covariance matrix with layout equal to that for `DF45#DAT`.

### 7.3.3 User-defined Random Regression

As a default, “`DXMRR`” fits a random regression on Legendre polynomials of the ‘meta-meter’, i.e. age or equivalent, to yield estimates of covariance functions as described by Kirkpatrick *et al.* (1990). In other case, the user may wish to fit another function. In this case, a formatted file `DF21#DAT` must be supplied which

has one record for each age (or equivalent) in the data, each record consisting of all the coefficients of the user-chosen function evaluated for that age. These must be column separated, REAL(8) variables which are read in a list-directed READ.

#### **7.3.4 Input of Starting Values from File**

For “DXMUX” and “DXMRR” starting values for the parameters to be estimated can optionally be read from a file rather than interactively from standard input. They should be given in a formatted file DF18#DAT, with one line per parameter, which again is read in a list-directed READ.

If a second animal effect is fitted and it is assumed to be uncorrelated to animals additive genetic effects, a value of 0.0 must be given as starting value for the first covariance between animal effects (e.g. direct-maternal genetic variance). If a value of zero is found for this component, the program will assume all other of the components (for other traits in a multivariate analysis) are zero as well.

#### **7.3.5 Multiple Measurement Error Variances**

For of “DXMRR”, where the number of measurement errors to be fitted is greater than one but less than the number of ages (or equivalent) found in the data, the program must be informed which measurement error variance is applicable to each age. This can be done interactively or by supplying a formatted file DF20#DAT. This file must contain one line per age, each containing an integer variable with the running number of measurement error variance appropriate for the age. Again, this is read in a list-directed READ.



---

## 8. Output Files in DFREML

---

### 8.1 Results File

For all programs and run options (except run option “8”), a formatted file `DF66#DAT` is written as the ‘results’ file, summarizing the data and pedigree structure, characteristics of the run performed, and giving estimates of (co)variance components, genetic parameters and effects fitted.

Under run option “8”, abbreviated results are written to `DF77#DAT`. This allows the user to check up on progress of ana analysis, by running a program with run option “8” while also running it with option “0”.

### 8.2 Other Output Files

#### 8.2.1 Monitoring Iterates

Under run option “0”, “`DxMUX`” and “`DxMRR`” write out a formatted file `Iterates` which allows the user to monitor convergence for the analysis. It accumulates the likelihood and norm of the parameter vector (AI-REML algorithm only) for each iterate, and is opened and closed for every iterate, so it can be inspected while the program is running (Unix environment).

#### 8.2.2 Parameter Estimates

“`DxMUX`” and “`DxMRR`” also write out formatted files with the parameter estimates at the end of iterations. This is called `DF19#DAT` for “`DxMUX`” (containing estimates of covariance components) and `DF17#DAT` for “`DxMRR`” (containing estimates of coefficient matrices of covariance functions).

If it is chosen to construct the covariance matrices for the ages in the data in “`DxMRR`”, it will write out a file `DF19#DAT` with these in addition. Furthermore, formatted files `siga`, `sigc`, `sigp`, `rra`, `rrc`, `rrp`, `hsq` and `sgpp` are written under this option, containing the constructed matrices of additive genetic covariances, permanent environmental covariances, phenotypic covariances, additive genetic correlations, permanent environmental correlations, phenotypic correlations, and the vectors of heritabilities and phenotypic standard deviations for the ages in the data. These are supplied to facilitate a quick graphic inspection of results.

#### 8.2.3 Breeding value estimates

Running “`DxMRR`” with run option “5”, two additional, formatted files may be written :

1. If a fixed regression on Legendre polynomials of age (or equivalent) has been fitted, “`DxMRR`” provides the option to evaluate estimated regression (GLS estimates at convergence) for the ages in the data. If chosen, a file `Orthog_reg_fix` is written out which contains one record per age, with variables : age, class mean, predicted least-squares mean (ignoring random effect) and predicted

generalised least-squares mean. This allows mean trajectories to be plotted readily.

2. “DXMRR” also provides the facility to calculate estimated breeding values for given age(s) from the estimated additive genetic, random regression coefficients for each animal. If this option is chosen, the breeding value estimates are written to a file `EBVs_for_ages`.

#### 8.2.4 ‘Re-constructed’ Covariance Matrices

Under run option “10”, “DXMRR” writes out a formatted file `DF15#DAT`. It contains one line for each pair of ages with the running numbers of the two ages, the actual values of the ages and the calculated covariance between them. The complete covariance matrix is written out (rather than just the upper or lower triangle), and ‘rows’ are separated by a blank line. `DF15#DAT` is directly suitable for plotting covariance functions, e.g. using ‘splot’ in `GNUPLOT`.

---

## 9. Examples

---

DFREML 3.0 comes with three worked examples. This allows the user to check that his/her installation is working correctly, and – hopefully – clarifies the steps involved in running a DFREML analysis.

Each example consists of some test data, files of ‘interactive input’ for various run options, and corresponding output files and screen logs; see Table 9.1 for a summary.

Table 9.1: Files for the worked examples

| Program          | Run Opt. | UNIX     |           |            | DOS        |             |              |
|------------------|----------|----------|-----------|------------|------------|-------------|--------------|
|                  |          | Input    | Output    | Results    | Input      | Output      | Results      |
| <b>Example 1</b> |          |          |           |            |            |             |              |
| “DFPREP”         |          | in.prep  | log.prep  | df66.prep  | prepuni.in | prepuni.out | prepuni.df66 |
| “DFUNI”          | -1       | in.uni1  | log.uni1  | df66.uni1  | uni1.in    | uni1.out    | uni1.df66    |
| “DFUNI”          | -2       | in.uni2  | log.uni2  | df66.uni2  | uni2.in    | uni2.out    | uni2.df66    |
| “DFUNI”          | 0        | in.uni0  | log.uni0  | df66.uni0  | uni0.in    | uni0.out    | uni0.df66    |
| “DFUNI”          | 5        | in.uni5  | log.uni5  | df66.uni5  | uni5.in    | uni5.out    | uni5.df66    |
| <b>Example 2</b> |          |          |           |            |            |             |              |
| “DFPREP”         |          | in.prep  | log.prep  | df66.prep  | prepmux.in | prepmux.out | prepmux.df66 |
| “DXMUX”          | -1       | in.mux1  | log.mux1  | df66.mux1  | mux1.in    | mux1.out    | mux1.df66    |
| “DXMUX”          | -2       | in.mux2  | log.mux2  | df66.mux2  | mux2.in    | mux2.out    | mux2.df66    |
| “DXMUX”          | 0        | in.mux0  | log.mux0  | df66.mux0  | mux0.in    | mux0.out    | mux0.df66    |
| “DXMUX”          | 5        | in.mux5  | log.mux5  | df66.mux5  | mux5.in    | mux5.out    | mux5.df66    |
| <b>Example 3</b> |          |          |           |            |            |             |              |
| “DFPREP”         |          | in.prep  | log.prep  | df66.prep  | prepmrr.in | prepmrr.out | prepmrr.df66 |
| “DXMRR”          | -1       | in.mrr1  | log.mrr1  | df66.mrr1  | mrr1.in    | mrr1.out    | mrr1.df66    |
| “DXMRR”          | -2       | in.mrr2  | log.mrr2  | df66.mrr2  | mrr2.in    | mrr2.out    | mrr2.df66    |
| “DXMRR”          | 0        | in.mrr0a | log.mrr0a | df66.mrr0a | mrr0a.in   | mrr0a.out   | mrr0a.df66   |
| “DXMRR”          | 0        | in.mrr0b | log.mrr0b | df66.mrr0b | mrr0b.in   | mrr0b.out   | mrr0b.df66   |
| “DXMRR”          | 0        | in.mrr0c | log.mrr0c | df66.mrr0c | mrr0c.in   | mrr0c.out   | mrr0c.df66   |
| “DXMRR”          | 5        | in.mrr5  | log.mrr5  | df66.mrr5  | mrr5.in    | mrr5.out    | mrr5.df66    |

### 9.1 Running the Examples

Instructions in the following assume

- that the appropriate path to the programs has been set (see chapter 3)
- that we are running from the directory for the example (i.e. containing all the input and test data and auxiliary files)
- that DFREML has not been run in this directory before, i.e. that there are no files DF59#DAT, DF49#DAT or MAXPARM in the directory!

Check that these assumptions hold if your example run fails.

### 9.1.1 DOS

First, set up the directory `examples` with subdirectories `Example1`, `Example2` and `Example3`. By default, this is a subdirectory of `DFREML`.

- Obtain the self-extracting archive `dfreml.v30.examples.exe` from the web/ftp site.
- Unpack it by double-clicking on its icon (Windows) or executing the file (DOS).

Each example requires a run of “DFPREP” as the first step; after changing to the appropriate example directory, this can be performed as `dfprep.exe < prepxyz.in` where `xyz` stands for `uni`, `mux` or `mrr` for Example 1, Example 2 and Example 3, respectively.

The second step for each analysis, is a run of the appropriate program (`dfuni` for Example 1, `dxmux` for Example 2 and `dxmrr` for Example 3), denoted by ‘program.exe’ in the following under run option -1. This can be carried out as : `program.exe < xyz1.in`

The third step, involves run option -2 : `program.exe < xyz2.in`,

and estimation is then carried out as : `program.exe < xyz0.in`

For Example 3, there are three estimation runs, i.e. replace 0 above by 0a, 0b, 0c, respectively. Before running “DXMRR”, copy the auxiliary file `Measure_error` to `DF20#DAT`.

Finally, backsolutions for fixed or random effects fitted can be obtained using run option 5 : `program.exe < xyz5.in`

### 9.1.2 UNIX

First extract the archive of examples :

1. Obtain `dfreml.v30.examples.tar.gz` from the the web/ftp site.
2. Uncompress the archive : `unzip dfreml.v30.examples.tar.gz`
3. Extract the example files : `tar xvf dfreml.v30.examples.tar`

This will create directories `Example1`, `Example2` and `Example3` as subdirectories of `DFREML/UNI`, `DFREML/MUX` and `DFREML/MRR`, respectively.

Each example requires a run of “DFPREP” as the first step; after changing to the appropriate example directory, this can be performed as `dfprep < in.prep`

The second step for each analysis, is a run of the appropriate program (`dfuni` for Example 1, `dxmux` for Example 2 and `dxmrr` for Example 3), denoted by ‘program’ in the following, under run option -1. This can be carried out as : `program < in.xyz1` where `xyz` stands for `uni`, `mux` or `mrr`.

The third step, involves run option -2 : `program < in.xyz2`,

and estimation is then carried out as : `program < in.xyz0`

For Example 3, there are three estimation runs, i.e. replace 0 above by 0a, 0b, 0c, respectively. Before running “DXMRR”, copy the auxiliary file `Measure_error` to `DF20#DAT`.

Finally, backsolutions for fixed or random effects fitted can be obtained using run option 5 : `program <in.xyz5`

## 9.2 Example 1

The first example shows a univariate analysis for a ‘simple’ animal model with animals’ additive genetic value as the only random effect using “DFUNI”.

The test data given is that used by Meyer (1989) to illustrate univariate REML estimation via a derivative-free algorithm. They are simulated records for a trait with phenotypic variance of 100, direct heritability of 0.40, maternal heritability of 0.20, maternal-direct covariance (divided by 100) of  $-0.05$  and a “c-squared” effect of 0.15. Data were generated for 2 generations of animals with a hierarchical full-sib family structure, yielding a total of 282 records and 306 animals in the analysis with generations as the only fixed effect.

Estimating only one genetic parameter (the heritability), the maximum of  $\log \mathcal{L}$  is located using a quadratic approximation of the likelihood.

## 9.3 Example 2

The second example shows a bivariate analysis under Model 2, i.e. fitting an additional, uncorrelated random effect, using an AI-REML algorithm and program “DxMUX”.

The test data is that used by Meyer (1991) to illustrate a multivariate, derivative-free REML algorithm. It consists of records for six week weight and adjusted food intake from 4 to 6 weeks for 3 generation of mice subject to within full-sib family selection for appetite. There are records for both traits for 284 animals from 42 litters. Fixed effects to be taken into account are generation, sex and litter size class.

## 9.4 Example 3

The third example shows a covariance function analysis using “DXMRR”. Again, the analysis is under Model 2, the additional random effect fitted being the permanent environmental effect due to the animal.

Test data given are 1626 January weights of 436 beef cows (from the Wokalup selection experiment in Western Australia), recorded between 19 and 82 months of age. There are 24 different ages (in months) in the data and up to 6 records per cow. The total number of animals in the analysis (including parents without records, after pruning) is 585. Analysis is carried out within 83 contemporary groups (year-paddock-age(in years) of weighing subclasses), fitted as fixed effects.

A polynomial order of fit of 3, i.e. a quadratic function, is chosen, and a corresponding fixed regression on Legendre polynomials of age is fitted. Six different measurement error variances, corresponding to the 6 years of age, are fitted. This yields a total of 18 parameters to be estimated. The analysis is carried out using both a derivative-free search and an AI-REML algorithm : First (a), 2 rounds of iteration using Powell’s derivative-free search are carried out. Second (b), estimation is carried out using an AI-REML algorithm. Third (c), convergence is checked.

---

## 10. References

---

- Edwards, A.W.F. 1961. *Likelihood*. Cambridge University Press.
- George, A. and Liu, J. W.-H. 1981. *Computer Solution of Large Positive Definite Systems*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632..
- 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.
- Graser, H.-U., Smith, S.P. and Tier, B. 1987. A derivative-free approach for estimating variance components in animal models by Restricted Maximum Likelihood.. *J. Anim. Sci.* **64** : 1362–1370.
- Harville, D.A. 1977. Maximum likelihood approaches to variance component estimation and to related problems. *J. Anim. Stat. Ass.* **72** : 320–338.
- 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. *J. Dairy Sci.* **78** : 449–456.
- Kirkpatrick, M., Lofsvold, D. and Bulmer, M. 1990. Analysis of the inheritance, selection and evolution of growth trajectories. *Genetics* **124** : 979–993.
- Meyer, K. 1988. DFREML - a set of programs to estimate variance components under an Individual Animal Model. *J. Dairy Sci.* **71** : Supplement 2, 33–34 (Abstract).
- Meyer, K. 1989. Restricted Maximum Likelihood to estimate variance components for animal models with several random effects using a derivative-free algorithm. *Genet. Select. Evol.* **21** : 317–340..
- Meyer, K. 1991. Estimating variances and covariances for multivariate Animal Models by Restricted Maximum Likelihood. *Genet. Select. Evol.* **23** : 67–83.
- Meyer, K. 1997. An “average information” Restricted Maximum Likelihood algorithm for estimating reduced rank genetic covariance matrices or covariance functions for animal models with equal design matrices. *Genet. Select. Evol.* **29** : 97–116.
- Meyer, K. and Smith, S.P. 1996. Restricted Maximum Likelihood estimation for animal models using derivatives of the likelihood . *Genet. Select. Evol.* **28** : 23–49.
- Meyer, K. and Hill, W.G. 1997. Estimation of genetic and phenotypic covariance functions for longitudinal or ‘repeated’ records by restricted maximum likelihood. *Livest. Prod. Sci.* **47** : 185–200.
- Nelder, J.A. and Mead, R. 1965. A simplex method for function minimization. *Computer J.* **7** : 147–151.
- Patterson, L.D. and Thompson, R. 1971. Recovery of inter-block information when block sizes are unequal. *Biometrika* **58** : 545–554.
- Powell, M.J.D. 1965. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer J.* **7** : 155–162.

- Quaas, R.L. 1976. Computing the diagonal elements of a large numerator relationship matrix. *Biometrics* **32** : 949–953.
- Smith, S.P. 1995. Differentiation of the Cholesky algorithm. *J. Comp. Graph. Stat.* **4** : 134–147.

---

# Index

---

- ]
- Covariance function, 13
  - Eigenfunction, 14
  - Eigenvalue, 14
- DFPREP, 10–12
  - Input files, 11
  - Interactive Input, 11
  - Tasks performed, 11
- DFUNI, 12–13
- DOS
  - Examples, 24
    - Download, 24
  - Programs
    - Download, 4
- Download
  - Programs
    - DOS, 4
    - UNIX, 4
- DXMRR, 13–14
- DXMUX, 13
- Error variance, 13
- Examples, 23
  - DOS, 24
  - Example 1, 25
  - Example 2, 25
  - Example 3, 25
  - UNIX, 24
- Installation
  - DOS, 4
  - UNIX, 4
    - Subroutine library, 4
- Longitudinal data, 13
- Mixed model matrix, 13
- Numerator relationship matrix, 12
- Orthogonal polynomials, 13
- Orthogonal polynomilas
  - Legendre, 13
- Pedigree



- Pruning, 12
- Recoding, 12
- Quadratic approximation, 13
- Random regression
  - Model, 13
  - Order of fit, 14
- Reduced rank, 14
- Repeated records, 13
- Right hand side
  - multiple, 13
- UNIX
  - Examples, 24
    - Download, 24
  - Programs
    - Download, 4
    - Installation, 4