ASReml Update

What's new in Release 2.00

A R Gilmour
NSW Department of Primary Industries, Orange, Australia

B R Cullis
NSW Department of Primary Industries, Wagga Wagga, Australia

S. A. Harding
VSN International, Hemel Hempsted, United Kingdom

R Thompson
Rothamsted Research, Harpenden, United Kingdom

ASReml Update. What's new in Release 2.00

A R Gilmour, B R Cullis, S A Harding and R Thompson

Published by:

VSN International Ltd,
5 The Waterhouse,
Waterhouse Street,
Hemel Hempstead,
HP1 1ES, UK
E-mail: info@asreml.co.uk
Website: http://www.vsni.co.uk/

Copyright Notice

The correct bibliographical reference for this document is:

Gilmour, A.R., Cullis, B.R., Harding, S. A. and Thompson, R. 2006 ASReml Update: What's new in Release 2.00 VSN International Ltd, Hemel Hempstead, HP1 1ES, UK

ISBN 1-904375-22-7

Author email addresses

Arthur.Gilmour@dpi.nsw.gov.au
Brian.Cullis@dpi.nsw.gov.au
Simon.Harding@vsni.co.uk
Robin.Thompson@bbsrc.ac.uk

Preface

ASReml, a statistical package that fits linear mixed models using Residual Maximum Likelihood (REML), is a joint project of NSW Department of Primary Industries[1] and Rothamsted Research[2]. It provides a stable platform for delivering well established procedures while also delivering current research in the application of linear mixed models. The strength of ASReml is the use of the Average Information (AI) algorithm and sparse matrix methods for fitting the linear mixed model. This enables it to analyse large and complex data sets quite efficiently.

This document highlights the developments in ASReml since Release 1.00 and is intended as a transition document for existing users. New users should refer to ASReml User Guide, Release 2.00.

Linear mixed effects models provide a rich and flexible tool for the analysis of many data sets commonly arising in the agricultural, biological, medical and environmental sciences. Typical applications include the analysis of (un)balanced longitudinal data, repeated measures analysis, the analysis of (un)balanced designed experiments, the analysis of multi-environment trials, the analysis of both univariate and multivariate animal breeding and genetics data and the analysis of regular or irregular spatial data.

ASReml is one of several user interfaces to the underlying computational engine. Genstat uses the same engine in its REML directive and the asreml class of functions available for S-Plus and R also use the same engine. Both of these have good data manipulation and graphical facilities.

The focus in developing ASReml has been on the core engine and it is freely acknowledged that its user interface is not to the level of these other packages. Nevertheless, as the developers interface, it is functional, it gives access to everything that the core can do and is especially suited to batch processing and running

[1] Kite St, Orange, New South Wales, 2800, Australia

[2] Harpenden, Hertsfordshire, United Kingdom

of large models without the overheads of other systems. Feedback from users is welcome and attempts will be made to rectify identified problems in ASReml .

Briefly, the improvements in Release 2.00 include more robust variance parameter updating so that 'Convergence Failure' is less likely, extensions to the syntax, improvements to the Analysis of Variance procedures, improvements to the handling of pedigrees and some increases in computational speed.

The data sets and ASReml input files used in this guide are available from `http://www.vsni.co.uk/products/asreml` as well as in the `examples` directory of the distribution CD-ROM.They remain the property of the authors or of the original source but may be freely distributed provided the source is acknowledged.

Proceeds from the licensing of ASReml are used to support continued development to implement new developments in the application of linear mixed models. The developmental version is available to supported licensees via a website upon request to VSN. Most users will not need to access the developmental version unless they are actively involved in testing a new development.

Acknowledgements

We gratefully acknowledge the Grains Research and Development Corporation of Australia for their financial support for our research since 1988. Brian Cullis and Arthur Gilmour thank the NSW Department of Primary Industries for providing a stimulating and exciting environment for applied biometrical research and consulting. Rothamsted Research receives grant-aided support from the Biotechnology and Biological Sciences Research Council of the United Kingdom.

We sincerely thank Ari Verbyla, Sue Welham, Dave Butler and Alison Smith, the other members of the ASReml 'team'. Ari contributed the cubic smoothing splines technology, information for the Marker map imputation, on-going testing of the software and numerous helpful discussions and insight. Sue Welham has overseen the incorporation of the core into Genstat and contributed to the `predict` functionality. Dave Butler has developed the asreml class of functions. Alison contributed to the development of many of the approaches for the analysis of multi-section trials. We also thank Ian White for his contribution to the spline methodology. The Matérn function material was developed with Kathy Haskard, a PhD student with Brian Cullis, and the denominator degrees of freedom material was developed with Sharon Nielsen, a Masters student with Brian Cullis. Damian Collins contributed the `PREDICT !PLOT` material. Greg Dutkowski has

contributed to the extended pedigree options. The `asremload.dll` functionality is provided under license to VSN. Alison Kelly has helped with the review of the XFA models. Finally, we especially thank our close associates who continually test the enhancements.

Arthur Gilmour acknowledges the grace of God through Jesus Christ our Saviour (*In Him are hidden all the treasures of wisdom and knowledge* Colossians 2:3)

Contents

1 System requirements

1.1 Introduction

These notes relating to ASReml 2.00 are intended for those familiar with release 1.00 and describe additions to the syntax. New users should read ASReml User Guide Release 2.00 which provides a comprehensive to fitting mixed models in ASReml .

1.2 Licensing

There are now two license models: individual license keys and network licenses. The former requires software registration followed by installation of the software using a procedure similar (but not identical) to ASReml 1.00. Use of a network key usually involves the cooperation of a network administrator for installation of the key and/or the software. See the installation instructions, provided with the software, for further information. Note that the new license keys permit use of any upgrades issued whilst support is maintained or during the period of an annual license.

1.3 Installation

ASReml 2.00 contains additional files added since Release 1.00, mainly providing more on-line documentation and user interfaces. Program files now incorporate the version number (e.g. asreml200.exe). This simplifies management of future updates, whilst using shortcuts (Windows) or symbolic links (Unix) to access the latest version. By default installation is to a new location, `asreml2`, to avoid conflict with existing copies.

Full installation instructions are provided separately.

1.4 User Interface

ASReml is essentially a batch program with some optional interactive features. The typical sequence of operations when using ASReml is

- Prepare the data (typically using a spreadsheet or data base program)
- Export that data as an ASCII file (for example export it as a `.csv` (comma separated values) file from Excel)
- Prepare a job file with filename extension `.as`
- Run the job file with ASReml
- Review the various output files
- revise the job and re-run it, or
- extract pertinent results for your report.

So you need an ASCII editor to prepare input files and review and print output files. We directly provide two options.

ASReml-W

ASReml-W is a graphical tool allowing the user to edit and run ASReml program files, and then view the output. It is available on the following platforms:

- Windows 32-bit,
- Windows 64-bit,
- Linux 32-bit,
- Linux 64-bit, and
- Sun/Solaris 32-bit

ASReml-W has a built-in help system explaining its use.

ConTEXT

ConTEXT is a third-party freeware text editor, with programming extensions which make it a suitable environment for running ASReml under Windows. The ConTEXT directory on the CD-ROM includes installation files and instructions for configuring it for use in ASReml . Full details of ConTEXT are available from `http://www.context.cx/`.

1.5 User Discussion List

An ASReml Discussion list is hosted by NSW Dept of Primary Industries. To join the list or change your email address,
request `arthur.gilmour@dpi.nsw.gov.au` to update the list. You may then direct your comments/queries to `ASREML-L@dpi.nsw.gov.au`.

1.6 Support

Help with installation, licensing and running ASReml is available to all users with a current support contract, by email to `support@asreml.co.uk`.

1.7 Tutorial

The distribution CD contains an ASReml tutorial in the form of sixteen sets of slides with audio (`.mp3`) discussion. The sessions last about 20 minutes each and should be taken in order over several days.

1.8 Help system

The ASReml help accessable through ASReml-W can also be accessed directly (`ASReml.chm`).

2 Critical Changes to Behaviour

Generally we seek to maintain upward compatability so that ASReml 1.00 code will continue to run. However, to deliver improved facilities, some changes to behaviour are unavoidable.

2.1 Workspace

ASReml will automatically increase workspace in large jobs if it needs it and it is available. The default workspace is 32Mbyte but the user can specify a larger amount (see Section 3.2). In any case, if a job runs out of workspace when it is running, and has not already claimed the maximum amount of workspace permitted on the computer, ASReml will restart the job with a larger allocation.

2.2 Storage of alphabetic factor labels

In release 1.00, there was a fixed allocation for alphabetic labels of 5000 labels of up to 20 characters each. Longer labels were truncated and regarded as equivalent if identical in the first 20 characters.

In release 2.00, the allocation is dynamic with default provision for 2000 labels of 16 characters each (see Section 5.1). However, the allocation is increased to make provision for the declared size of all alphabetic factors and the label length can be set using the `!LL` qualifier. If the number of levels is overspecified, the `!PRUNE` qualifier may be used to readjust the sizes.

2.3 Slash operator in the model specificaton

In model specification, `A/B` now expands to `A A.B`

2.4 Graphics

In this version, graphics have been converted from Interacter to Winteracter except that MENU mode has been removed, being replaced by ASReml-W. This should not alter the appearance of any of the graphs produced.

2.5 Singularities in Average Information matrix

Singularities in ASReml arise in three contexts: the linear model, the variance model and the Average Information (AI) matrix. The AI matrix is used to give updates to the variance parameter estimates. The AI matrix is used to give updates to the variance parameter estimates. In release 1.00, if singularities were present in the AI matrix, a generalized inverse was used which effectively conditioned on whichever parameters were identified as singular. ASReml now aborts processing if singularities appear unless the `!AISINGULARITIES` qualifier is set. Which particular parameter is singular is reported in the variance component table printed in the `.asr` file.

The most common reason for singularities is that the user has overspecified the model and is likely to misinterpret the results if not fully aware of the situation. Overspecification will occur in a direct product of two unconstrained variance matrices, when a random term is confounded with a fixed term and when there is no information in the data on a particular component. The best action is to reform the variance model so that the ambiguity is removed, or to fix one of the parameters in the variance model so that the model can be fitted. Only rarely will it be reasonable to specifiy the `!AISINGULARITIES` qualifier.

2.6 Prediction

The order predicted values are presented has changed and are now under user control: the factors are presented in the order they are declared on the `PREDICT` statement. The predicted values are ordered so that the rightmost factor rotates fastest, followed by the second from the right.

2.7 Extended Factor Analytic

XFA model fitting with some specific variances zero is extended to allow direct product structures of XFA with other structures (previously only XFA with Identity was allowed) provided the other structure is not another XFA with some zero specific variances, and to allow several model terms to have XFA structures involving zero specific variances. The work around for XFA x XFA (both with zero PSIs) is to fix 'ZERO' PSIs in one of the terms at 0.0001. In 3-way terms involving an XFA with zero specific variance, the central component must be an Identity. I.e. XFA $\otimes \boldsymbol{I} \otimes \boldsymbol{C}$ and $\boldsymbol{C} \otimes \boldsymbol{I} \otimes$ XFA are allowed where $\boldsymbol{C}$ is some structure and $\boldsymbol{I}$ is the Identity.

2.8 ASReml **update changes**

Changes to the rules for when the AI update of a variance parameter is replaced by a smaller update have been made to make the process more robust. Consequently, the iteration sequence may differ slightly in some jobs from earlier versions. It should end up at effectively the same point with only small numerical differences in the results. These rules come into play when the simple AI update is large (10 fold) or would put the parameter 'out of bounds'.

2.9 BLUEs in `.asr` file

In release 1.00, the non-zero BLUEs (estimates of fixed effects) were reported in the `.asr` file as well as in the `.sln` file. In release 2.00, the BLUEs are not reported in the `.asr` file unless the `!BRIEF -1` qualifier is set or there are less than 10 BLUEs to report.

3 Command Line

3.1 Template from data file

The facility to generate a template `.as` file has been moved from the `MENU` mode to the command line, and extended. Normally, the name of a `.as` command file is specified on the command line. If a `.as` file does not exist and a file with file extension `.asd`, `.csv`, `.dat`, `.gsh`, `.txt` or `.xls` is specified, ASReml assumes the data file has field labels in the first row and generates a `.as` file template. First, it seeks to convert the `.gsh` (Genstat) or `.xls` (Excel) file to `.csv` format using the `ASRemload.dll` utility provided by VSN (see page 19). In generating the `.as` template, ASReml takes the first line of the `.csv` (or other) file as providing column headings, and generates field definition lines from them. If some labels have `!` appended, these are defined as factors, otherwise ASReml attempts to identify factors from the field contents. The template needs further editing before it is ready to run but does have the field names copied across.

3.2 Command line Options

Command line options (some with arguments) are presented as a single concatenated string with a leading `-` as the first program argument. Remember that [ ] in the guide is used to indicate optional input and such square braces are not to be typed into the command file.

`A` (ASK) has been added to make it easier to specify command line options in Windows Explorer. One of the options available when right clicking a `.as` file, invokes ASReml with this option. ASReml then prompts for an option string and arguments string, allowing these to be set interactively at run time.

`B[`*b*`]` (BRIEF) suppresses some of the information written to the `.asr` file. The data summary and regression coefficient estimates are suppressed by the options B, B1 or B2. This option should not be used for initial runs of a job before you have confirmed (by checking the data summary) that ASReml has read the data as you intended. Use B2 to also have the predicted values written to the .asr file instead of the .pvs file. Use B-1 to get BLUE estimates reported in .asr file.

H[*g*] (HARDCOPY) has been added to replace the G option when graphics are to be written to file but not displayed on the screen. The H may be followed by a format code e.g. H22 for .eps.

J (JOIN) is used in association with the !CYCLE qualifier to put the output from a set of runs into single files (see !CYCLE list !JOIN in Chapter 4.2).

Q (QUIET) is used when running under the control of ASReml-W to suppress any POPUPs/ PAUSES from ASReml.

O (ONERUN) is used with the R option to make ASReml perform a single analysis when the R option would otherwise attempt multiple analyses. The R option then builds some arguments into the output file name while other arguments are not. For example
`ASReml -nor2 mabphen 2 TWT out(621) out(929)`
results in one run with output files mabphen2_TWT.*.

R[*r*] (REPEAT) is extended to allow an argument *r* with default value 1. R*r* is used in conjunction with at least *r* argument(s) and does two things: it modifies the output filename to include the first *r* arguments so the output is identified by these arguments, and, if there are more than *r* arguments, the job is rerun moving the extra arguments up to position *r* (unless ONERUN (O) is also set).

For example

```
ASReml -r2 job wwt gfw fd fat
```

is equivalent to running three jobs:

```
ASReml -r2 job wwt gfw → jobwwt_gfw.asr
ASReml -r2 job wwt fd → jobwwt_fd.asr
ASReml -r2 job wwt fat → jobwwt_fat.asr
```

W*m* (WORKSPACE) sets the initial size of the workspace. The default workspace (if W*m* is not specified) is 32Mbyte. For example W1600 requests 1600 Mbytes of workspace, the maximum typically available under Windows.

If the allocated amount is found to be inadequate, ASReml will attempt to resize it and restart the job. W2000 is the maximum available on 32bit Unix(Linux) systems. On 64bit systems, the argument, if less than 32, is taken as Gbyte.

If your system cannot provide the requested workspace, the request will be diminished until it can be satisfied. On multi-user systems, do not unnecessarily request the maximum or other users may complain.

The workspace requirements depend on problem size and may be quite large. If the allocated amount is found to be inadequate, ASReml will attempt to resize it and restart the job.

3.3 Job control line, a new optional header line.

Since Windows likes to hide the command line, most command line options can be set on an optional new initial line of the `.as` file. If the first line of the `.as` file contains a qualifier other than `!DOPATH`, it is interpreted as setting command line options and the <Title> is taken as the next line.

The option string actually used by ASReml is the combination of what is on the command line and what is on the job control line, with options set in both places taking arguments from the command line. Arguments on the job control line are ignored if there are arguments on the command line.

The options can be set either as a concatenated string in the same format as expected on the command line, or as a list of qualifiers. In the former case, the syntax is

`!-`*s* *a*

where *s* is the option letter string as defined for the command line options and *a* is a list of command line arguments. For example

```
!-h22r 1 2 3
```

on the first line is equivalent to running ASReml with the command line

```
ASReml -h22r jobname 1 2 3
```

Alternatively, `!ASK` prompts for an options string and arguments (like the `A` command line option). It is assumed that no other qualifiers are set on this line when `!ASK` is specified. For example

```
-h22r 1 2 3
```

might be the response. The allowed options are -B*b*CDEFG*g*H*g*IJLNOR*r*S*s*W*w*Y*y*

The qualifiers to individually specify command line options are as follows.

`!ARGS` a	rest of line taken as command line arguments (a) This qualifier must be specified last on the line.
`!BRIEF` b	same as `B`b
`!CONTINUE`	same as `C`
`!DEBUG`	same as `D`
`!DEBUG 2`	same as `DE`
`!FINAL`	same as `F`
`!GRAPHICS` g	same as `G`g
`!HARDCOPY` g	same as `H`g
`!LOG`	same as `L`
`!JOIN`	same as `J`
`!NOGRAPHS`	same as `N`
`!ONERUN`	same as `O`
`!QUIET`	same as `Q`
`!REPEAT` r	same as `R`r
`!WORK` w	same as `W`w
`!Y` y	same as `Y`y

The following additional qualifiers provide an alternative to using a numeric argument on the `GRAPHICS` or `HARDCOPY` qualifiers to set the type of graphics file produced.

`!BMP`	sets graphics device to BitMap ($g = 6$)
`!EPS`	sets graphics device to Encapsulated PostScript ($g = 22$)
`!HPGL`	sets graphics device to HP GL ($g = 1$)
`!HPGL 12`	sets graphics device to HP GL 2($g = 12$)
`!PS`	sets graphics device to PostScript ($g = 2$)
`!WMF`	sets graphics device to Windows Meta File ($g = 11$)

4 Paths and loops in the .as file

ASReml is designed to analyse just one model per run. However, the analysis of a data set typically requires many runs, fitting different models to different traits. It is often convenient to have all these runs coded into a single .as file and control the details from the command line (or job control line) using arguments.

4.1 Paths

Which particular lines in the .as file are honoured is primarily controlled by the !DOPATH (or !DOPART) qualifier in conjunction with !PATH (or !PART) statements.

!DOPATH *i* can (now) be located anywhere in the job, including on the first or second line. The argument (*i*) controls which lines (delineated by !PATH statements) are honoured in the particular run. Often, the argument (*i*) is given as $1 indicating that the actual path to use is specified as the first argument on the command/job control line. If !DOPATH $1 is placed on the job control line, the arguments must be supplied on the command line rather than on the job control line.

The !PATH (or !PART) control statement may list multiple path numbers so that the following lines are honoured if any one of the listed path numbers is active. This qualifier must appear at the beginning of its own line anywhere after the !DOPATH qualifier. For example

```
shf.dat !DOPART 4
!PATH 2 4 6
```

4.2 Loops

`!CYCLE` <list> `!JOIN`
is a mechanism whereby ASReml can loop through a series of jobs, writing the output to separate files or a single file if `!JOIN` specified. The `!CYCLE` qualifier must appear on its own line anywhere in the job, starting in character 1; it can occur before the 'title' line (but after the job control line if it is present). If <list> has n values, the job is run n times with the ith value substituted into the job everywhere that the `$I` string appears. The value is also built into the output filename if `!JOIN` is omitted. For example

```
!CYCLE 0.4 0.5 0.6 !JOIN
20 0 mat2 1.9 $I !GPF
```

would result in three runs and the results would be appended to a single file.

5 Field Definition qualifiers

5.1 Storage of alphabetic factor labels

The storage of factor level labels has changed. Previously there was space for 5000 labels of 20 characters each. Now space is allocated dynamically with default allocation being 2000 labels of 16 characters long. If there are large `!A` factors (so that the total across all factors will exceed 2000), you must specify the anticipated size (within say 5%). If some labels are longer then 16 characters and the extra characters are significant, you must lengthen the space for each label by specifying `!LL` *c* e.g.

```
cross !A 2300 !LL 48
```

indicates the factor `cross` will have about 2300 levels and needs 48 characters to hold the level names. Note that only the first 20 characters of the labels are ever printed.

`!PRUNE` on a field definition line means that if fewer levels are actually present in the factor than were declared, ASReml will reduce the factor size to the actual number of levels. Use `!PRUNALL` for this action to be taken on the current and subsequent factors up to (but not including) a factor with the `!PRUNEOFF` qualifier. The user may overestimate the size for large ALPHA and INTEGER coded factors so that ASReml reserves enough space for the list. Using `!PRUNE` will mean the extra (undefined) levels will not appear in the `.sln` file. Since it is sometimes necessary that factors not be pruned in this way, for example in pedigree/GIV factors, pruning is only done if requested.

5.2 Reordering the factor levels

`!SORT` declared after `!A` or `!I` on a field definition line will cause ASReml to sort the levels so that labels occur in alphabetic/numeric order for the analysis. By default, ASReml orders factor levels in the order they appear in the data so that for example, the user cannot tell whether `SEX` will be coded `1=Male, 2=Female` or `1=Female, 2=Male` without looking at the data file to see whether `Male` or

`Female` appears first in the `SEX` field. With the `!SORT` qualifier, the coding will be `1=Female, 2=Male` regardless of which appears first in the file.

`!SORTALL` means that the levels for the current and subsequent factors are to be sorted.

5.3 Skipping input fields

`!SKIP` *f* will skip *f* data fields BEFORE reading this field. It is particularly useful in large files with alphabetic fields which are not needed as it saves ASReml the time required to classify the alphabetic labels. For example

```
    Sire !I !skip 1
```

would skip the field before the field which is read as 'Sire'.

5.4 Reading date fields

`!DATE` indicates the field has one of the date formats dd/mm/yy, dd/mm/ccyy, dd-Mon-yy, dd-Mon-ccyy, hh:mm:ss and is to be converted into a Julian day or seconds past midnight where dd is a 1 or 2 digit day of the month, mm is a 1 or 2 digit month of the year, Mon is a three letter month name (`Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec`) yy is the year within the century (00 to 99), cc is the century (18, 19 or 20), hh is hours (1 to 23), mm is minutes (0-59) and ss is seconds (0 to 59). The separators '/', '-' and ':' must be present as indicated. The dates are converted to days since 31st December 1899. When the century is not specified, yy of 0-32 is taken as 2000-2032, 33-99 taken as 1933-1999.

`!DMY` indicates the field has one of the date formats dd/mm/yy or dd/mm/ccyy and is to be converted into a Julian day.

`!MDY` indicates the field has one of the date formats mm/dd/yy or mm/dd/ccyy and is to be converted into a Julian day.

5.5 New transformations

`!D` *v* (drop records with *v* or *missing* in the field) has been extended to allow relational operators `<`, `<=`, `<>`, `>=` and `>` to be inserted before *v*.

`!M` v (change the data value v to *missing* in the field) has been extended to allow relational operators `<`, `<=`, `<>`, `>=` and `>` to be inserted before v.

`!NORMAL` v replaces the variate with normal random variables having variance v.

For example, `Ndat !=0. !Normal 4.5` creates a new variable (`!=0.`) and fills it with Normal(0,4.5) random values. These two transformations can be collapsed into one: viz.

```
Ndat !=Normal 4.5
```

`!REPLACE` o n replaces data values o with n in the current variable. I.e.

```
    IF(DataValue.EQ.o) DataValue=n
```

`!RESCALE` o s rescales the column(s) in the current variable (`!G` group of variables) using $\boldsymbol{Y} = (\boldsymbol{Y} + o) * s$

`!SEED` n sets the seed for the random number generator. For example, `!SEED 848586` sets the seed for the random number generator to 848586.

`!SETN` v n replaces data values $1 : n$ with normal random variables having variance v. Data values outside the range $1 \cdots n$ are set to 0. For example, `Anorm !=A !SETN 2.5 10` replaces data values of 1, $\cdots$, 10 (copied from variable `A`) with 10 Normal(0,2.5) random values.

`!SETU` v n replaces data values $1 : n$ with uniform random variables having range $0 : v$. Data values outside the range $1 \cdots n$ are set to 0. For example, `Aeff !=A !SETU 5 10` replaces data values of 1, $\cdots$, 10 (copied from variable `A`) with 10 Uniform(0,5) random values.

`!TIME` indicates the field has the time format *hh:mm:ss* and is to be converted into seconds past midnight where hh is hours (1 to 23), mm is minutes (0-59) and ss is seconds (0 to 59). The separator ':' must be present as indicated.

`!UNIFORM` v replaces the variate with uniform random variables having range $0 : v$. For example,

```
 Udat !=0.  !Uniform 4.5
```

creates a new variable (`!=0.`) and fills it with Uniform(0,4.5) random values. These two transformations can be collapsed into one: viz.

```
Udat !=Uniform 4.5
```

`!MM` s associates marker positions in the vector s (based on the Haldane mapping function) with marker variables and replaces missing values in a vector of marker states with expected values calculated using distances to non-missing flanking markers. This transformation will normally be used on a `!G` n factor where the n variables are the marker states for n markers in a linkage group in map order and coded [-1,1] (backcross) or [-1,0,1] (F2 design). s (length n+1) should be the n marker positions relative to a left telomere position of zero, and an extra value being the length of the linkage group (the position of the right telomere). The length (right telomere) may be omitted in which case the last marker is taken as the end of the linkage group. The positions may be given in Morgans or centiMorgans (if the length is greater than 10, it will be divided by 100 to convert to Morgans).

The recombination rate between markers at s_L and s_R (L is left and R is right of some putative QTL at Q) is
$\theta_{LR} = (1 - e^{-2(s_R - s_L)})/2$.
Consequently, for 3 markers (L,Q,R), $\theta_{LR} = \theta_{LQ} + \theta_{QR} - 2\theta_{LQ}\theta_{QR}$.
The expected value of a missing marker at Q (between L and R) depends on the marker states at L and R: $E(q|1,1) = (1 - \theta_{LQ} - \theta_{QR})/(1 - \theta_{LR})$,
$E(q|1,-1) = (\theta_{QR} - \theta_{LQ})/\theta_{LR}$, $E(q|-1,1) = (\theta_{LQ} - \theta_{QR})/\theta_{LR}$
and $E(q|-1,-1) = (-1 + \theta_{LQ} + \theta_{QR})/(1 - \theta_{LR})$.
Let $\lambda_L = (E(q|1,1) + E(q|1,-1))/2 = \frac{\theta_{QR}(1-\theta_{QR})(1-2\theta_{LQ})}{\theta_{LR}(1-\theta_{LR})}$
and $\lambda_R = (E(q|-1,1) + E(q|-1,-1))/2 = \frac{\theta_{LQ}(1-\theta_{LQ})(1-2\theta_{QR})}{\theta_{LR}(1-\theta_{LR})}$
Then $E(q|x_L, x_R) = \lambda_L x_L + \lambda_R x_R$. Where there is no marker on one side, $E(q|x_R) = (1 - \theta_{QR})x_R + \theta_{QR}(-x_R) = x_R(1 - 2\theta_{QR})$.

`!DOM` A is used to form dominance covariables from a set of additive marker covariables previously declared with the`!MM` marker map qualifier. It assumes the argument A is an existing group of marker variables relating to a linkage group defined using `!MM` which represents additive marker variation coded [-1, 0, 1] (representing marker states aa, aA and AA) respectively. It is a group transformation which takes the [-1,1] interval values, and calculates $(|X| - 0.5) * 2$ i.e. -1 and 1 become one, 0 becomes -1. The marker map is also copied and applied to this model term so it can be the argument in a `qtl()` term (page 37).

6 Pedigree and GIV files

6.1 GIV Files

The standard .giv file procedure expects the user will supply an inverse matrix. In some situations, it is easier to form the uninverted matrix and not very convenient for the user to invert it outside of ASReml to create the .giv file. In this case, supply the uninverted matrix in the sparse format file but with a file extension .grm. ASReml will then invert the matrix itself before it uses it.

6.2 Pedigree file line qualifiers

Formation of the A-inverse has been speeded up (substantial gain if many animals without progeny)

Some new pedigree processing options added are:[1]

!MGS now formed directly rather than by inserting dummy DAMs.

!SELF s allows partial selfing when 'Dam'=='Male parent' unspecified. . It indicates that progeny from a cross where the second parent (male_parent) is unknown, is assumed to be selfed with probability s and outcrossed with probability $(1 - s)$.[2] This is appropriate in some forestry tree breeding studies where seed collected from a tree may be partly selfed and partly open pollinated. Do not use the !SELF qualifier with the !INBRED or !MGS qualifiers.

!INBRED v generates pedigree for inbred lines.

[1] A white paper downloadable from http://www.vsni.co.uk/resources/doc/ contains details of these options.

[2] Dutkowski GW, Gilmour AR (2001). Modification of the additive relationship matrix for open pollinated trials. In 'Developing the Eucalypt of the Future'. 10-15 September, Valdivia, Chile. p. 71. (Instituto Forestal: Chile)

Each cross is assumed to be selfed several times to stabilize as an inbred line as is usual for cereals, before being evaluated or crossed with another line. The argument v has default value of 1. and is the inbreeding coefficient for 'base' individuals. Caution Since inbreeding is usually associated with strong selection, it is not obvious that a pedigree assumption of covariance of 0.5 between parent and offspring actually holds.[3] Do not use the `!INBRED` qualifier with the `!MGS` or `!SELF` qualifiers.

The `!DIAG` qualifier used to return the diagonal of the A-inverse matrix in `AINVERSE.DIA`. Now it also returns the inbreeding coefficients for the individuals in this file (calculated as the diagonal of $\boldsymbol{A} - \boldsymbol{I}$).

`!SORT` causes ASReml to sort the pedigree into an acceptable order, that is parents before offspring, before forming the A-Inverse. The sorted pedigree is written to a file whose name has `.srt` appended to it. ASReml then forms the A-inverse from this new file.

[3] There is possibly the need for a variation on this theme where base individuals are assumed inbred but the data is collected on crossbred individuals.

7 Reading Data

7.1 Preparing data files in Excel

Many users find it convenient to prepare their data in Excel or Access. However, the data must be exported from these programs in either `.csv` (Comma separated values) or `.txt` (TAB separated values) form for `.asr`to read it. Care must be taken with missing values which commonly appear as empty fields, `NA`, `*` or `..` `.asr`will not recognise empty fields except in `.csv` files.

`.asr`has a facility to convert an `.xls` file to a `.csv` file. It is invoked if there is no `.csv` file or `.as` with the same basename (see page 7). It will also convert a Genstat `.gsh` spreadsheet file to `.csv` format. The data extracted are labels, numerical values and the results from formulae. A label of * in an otherwise numerical column is taken as a missing value as are empty cells. Empty rows at the start and end of a block are trimmed, but empty rows in the middle of a block are kept. Empty columns are also ignored. A single row of labels as the first non-empty row in the block will be taken as column names. Empty cells in this row will have a default names C1, C2 etc. assigned.

7.2 Combining columns from separate files

`!MERGE` c <filename> [`!SKIP` n] [`!MATCH` a b]
qualifiers may be specified on a line following the data filename line. The purpose is to combine data fields from the (primary) data file with data fields from the secondary (`!MERGE`). The effect is to open the named file (skip n lines) and then insert the columns from the new file into field positions starting at position c. If `!MATCH` a b is specified, ASReml checks that the field a $(0 < a < c)$ has the same value as field b. If not, it is assumed that the merged file has some missing records and missing values are inserted into the data record and the line from the MERGE file is kept for comparison with the next record. At this stage it is expected that the lines in the MERGE file are in the same order as the corresponding lines occur in the primary data file, and that there are no

extraneous lines in the MERGE file.[1]

For example, assuming the field definitions define 10 fields,

```
 PRIMARY.DAT   !skip 1
  !MERGE 6 SECOND.DAT  !SKIP 1  !MATCH 1 6
```

would obtain the first five fields from PRIMARY.DAT and the next five from SECOND.DAT, checking that the first field in each file has the same value.

Thus each input record is obtained by combining information from each file, before any transformations are performed.

7.3 Combining rows from separate files

ASReml can read data from multiple files provided the files have the same layout. The file specified as the data file can contain lines of the form
`!INCLUDE` <filename> `!SKIP` n
where <filename> is the (path)name of the data subfile and `!SKIP` n is an optional qualifier indicating that the first n lines of the subfile are to be skipped. Typically, the primary data file will just contain `!INCLUDE` statements identifying the subfiles to include. For example, you may have data from a series of related experiments in separate data files for individual analysis. The data file for the subsequent combined analysis would then just contain a set of `!INCLUDE` statements to specify which experiments were being combined.

After reading each subfile, input reverts to the primary data file.

If the subfiles have CSV format, they should all have it and the `!CSV` file should be declared on the primary datafile line. This option is not available in combination with `!MERGE`.

[1] It is proposed to extend this so the orders do not need to agree and that multiple lines in the primary file could be merged with the same line of the MERGE file.

8 Data file line qualifiers

Datafile line qualifiers may also be defined using an environment variable called `ASREML_QUAL`. The environment variable is processed immediately after the data file name line is processed. All qualifier settings are reported in the `.asr` file.

`!AILOADINGS` *i* controls modification to AI updates of loadings in factor analytic variance models. After ASReml calculates updates for variance parameters, it checks whether the updates are reasonable and sometimes reduces them. For factor loadings, the default behaviour is to shrink the loadings only in the first iteration if they appear large. This qualifier gives some user control. If it is specified without an argument, no (extra) shrinkage is allowed. Otherwise shrinkage is allowed in the first *i* iterations.

`!AISINGULARITIES` can be specified to force a job to continue even though a singularity was detected in the AI matrix. In release 1.00, if singularities were present in the Average Information matrix, a generalized inverse was used which effectively conditioned on whichever parameters were identified as singular. ASReml now aborts processing if singularities appear unless the `!AISINGULARITIES` qualifier is set. Which particular parameter is singular is reported in the variance component table printed in the `.asr` file.

The most common reason for singularities is that the user has overspecified the model and is likely to misinterpret the results if not fully aware of the situation. Overspecification will occur in a direct product of two unconstrained variance matrices, when a random term is confounded with a fixed term and when there is no information in the data on a particular component. The best action is to reform the variance model so that the ambiguity is removed, or to fix one of the parameters in the variance model so that the model can be fitted. Only rarely will it be reasonable to specify the `!AISINGULARITIES` qualifier.

`!BMP` sets hardcopy graphics file type to `.bmp`.

`!BRIEF` suppresses some of the information written to the `.asr` file. The data summary and regression coefficient estimates are suppressed. This qualifier should not be used for initial runs of a job until the user has confirmed from the data summary that the data is correctly interpreted by ASReml. Use `!BRIEF` 2 to cause the predicted values to be written to the `.asr` file instead of the `.pvs` file. Use `!BRIEF` -1 to get BLUE estimates reported in `.asr` file. The `!BRIEF` qualifier may be set with the `B` command line option.

`!CONTRAST` <label> <ref> <values>
provides a convenient way to define contrasts among treatment levels. `!CONTRAST` lines occur between the Data File Name line and the model line.

<label> is the name of the model term being defined.

<ref> is the name of an existing factor.

<values> is the list of contrast coefficients. For example

```
!CONTRAST LinN Nitrogen 3 1 -1 -3
```

defines `LinN` as a contrast based on the 4 (implied by the length of the list) levels of factor `Nitrogen`. The user should check the levels of the factor are in the order assumed by contrast (check the `.ass` or `.sln` or `.tab` files). Missing values in the factor become missing values in the contrast. Zero values in the factor (no level assigned) become zeros in the contrast.

`!DDF` [*i*] controls the calculation of denominator degrees of freedom required to calculate the significance of F statistics in the Analysis of Variance. There are three options:
`!DDF` -1 suppresses calculation of Denominator DF. Since calculation of the denominator degrees of freedom is computationally expensive, use `!DDF -1` when there is no interest in performing significance testing of the F statistics.
`!DDF 1` and `!DDF [2]` use different methods of calculating the derivatives of the inverse coefficient matrix but otherwise both use the formulas of Kenward and Roger (1997) to calculate the denominator degrees of freedom. `!DDF 1` uses numerical derivatives. This effectively requires an extra evaluation of the mixed model equations for every variance parameter. Consequently, calculating the derivatives this way can easily double the execution time for the job. `!DDF [2]` calculates the derivatives algebraically but this requires forming a large dense matrix, potentially of order number of equations plus number of records. It therefore can easily run out of space.

The default is to use algebraic derivatives if the qualifier is given with no argument or the qualifier is omitted and the job is relatively small. ($<$ 10 parameters, $<$ 500 fixed effects, $< 10,000$ equations and $<$ 100 Mbyte workspace). Algebraic derivatives are not available when MAXIT is 1, for multivariate analysis and for jobs with more than 10,000 equations.

`!DENSE` [n] has been modified to accept an argument up to 5000. The upper limit in release 1.00 was 800 which is still the default.

`!EPS` sets hardcopy graphics file type to `.eps`.

`!EQORDER` o modifies the algorithm used for choosing the order for solving the mixed model equations. A new algorithm devised for release 2.00 is now the default and is formally selected by `!EQORDER 3`. The algorithm used for release 1.00 is essentially that selected by `!EQORDER 1`. The new order is generally superior. `!EQORDER -1` instructs ASReml to process the equations in the order they are specified in the model. Generally this is disastrous but in one particular case is advantageous. It is the case where the model is specified as

```
      Y ~ mu !r !{ giv(id) id !}
```

and `giv(id)` invokes a dense inverse of an IBD matrix and `id` has a sparse structured inverse of an additive relationship matrix. While `!EQORDER 3` generates a more sparse solution, `!EQORDER -1` is faster to solve in this case.

`!FCON` adds a 'conditional' F-statistic column to the Analysis of Variance table. This conditional F-statistic tests each model term as if it was fitted as far down the table as possible but before other terms of which it is a component. The detail of exactly which terms are omitted is reported in the `.aov` file. The principle used in determining this conditional test is that a term cannot be adjusted for another term which encompasses it explicitly (e.g. term `A.C` cannot be adjusted for `A.B.C`) or implicitly (e.g. term `REGION` cannot be adjusted for `LOCATION` when locations are actually nested in regions although they are coded independently). See a separate full discussion of the new Analysis of Variance.

`!GKRIGE` [p] controls the expansion of `!PVAL` lists for `fac(`X`,`Y`)` model terms. For kriging prediction in 2 dimensions (X,Y), typically the user will want to predict at a grid of values, not necessarily just at data combinations. The values at which the prediction is required can be specified separately for X and Y using `!PVAL` statements. Normally, predict points will be defined for all combinations of X and Y values. This qualifier is required (with optional argument `1`) to specify the lists are to be taken in parallel. The lists must be the same length if to be taken in parallel.

`!HPGL [2]` sets hardcopy graphics file type to HP GL. An argument of 2 sets the hardcopy graphics file type to HP GL 2

`!LAST` <factor$_1$ > <lev$_1$ > [<fac$_2$ > <lev$_2$ > <fac$_3$ > <lev$_3$ >]
limits the order in which equations are solved in ASReml by forcing equations in

the sparse partition involving the the first <lev$_i$ > equations of <factor$_i$ > to be solved after all other equations in the sparse partition. Is intended for use when there are multiple fixed terms in the sparse equations so that ASReml will be consistent in which effects are identified as singular. The test example had

```
!r Anim Litter !f HYS
```

where genetic groups were included in the definition of `Anim`. Consequently, there were 5 singularities in `Anim`. The default reordering allows those singularities to appear anywhere in the `Anim` and `HYS` terms. In the more general model fitting

```
!r Tr.Anim Tr.Lit !f Tr.HYS
```

the location of singularities will almost surely change if the G structures for `Tr.Anim` or `Tr.Lit` are changed, invalidating Likelihood Ratio tests between the models. Since 29 genetic groups were defined in `Anim`, `!LAST Anim 29` forces the genetic group equations to be absorbed last (and therefore incorporate any singularities).

`!MBF mbf(<X>,`*m*`) <filename> !SKIP` *n*
specified on a separate line after the data file name line predefines the model term `mbf(<X>,`*m*`)` as a set of *m* covariates indexed by the data values. MBF stands for My Basis Function and uses the same mechanism as the `leg(), pol()` and `spl()` model functions but with covariates supplied by the user. <X> is the variate in the data containing the data values to be used to associate the covariate values with the records. *m* is the number of covariates to be obtained from the file <filename>. That is, <filename> is a file containing 1+*m* fields where the first field contains the data values, the remaining *m* fields define the corresponding covariate valies. If prediction is required at values of <X> not present in the data, the file should also include lines for these extra prediction points. `!SKIP` *n* is an optional qualifier which requests the first *n* lines of the file be ignored.

`!PS` sets hardcopy graphics file type to `.ps`.

`!PVSFORM` *f* modifies the format of the tables in the `.pvs` file and changes the file extension of the file to reflect the format.
`!PVSFORM` 1 means TAB separated: `.pvs` becomes `_pvs.txt`
`!PVSFORM` 2 means COMMA separated: `.pvs` becomes `_pvs.csv`
`!PVSFORM` 3 means Ampersand separated: `.pvs` becomes `_pvs.tex`
See `!TXTFORM` for more detail.

`!RREC` [*n*] causes ASReml to read *n* records or to read up to a data reading error if *n* is omitted, and then process the records it has. This allows data to be extracted from a file which contains trailing non-data records (for example extracting the predicted values from a `.pvs` file). The argument (*n*) specifies the number of data records to be read. If n is not supplied, ASReml reads until a data reading error occurs, and then processes the data it has. Without this qualifier, ASReml aborts the job when it encounters a data error. See `!RSKIP`.

`!RSKIP` n [s] instructs ASReml to skip the leading lines of the data file up to and including the nth instance of the character string s (default value `Ecode`). For example

```
!RREC !RSKIP 3 ' Ecode'
```

would enable ASReml to read the second block of predicted values from a `.pvs` file since the string `Ecode` occurs once at the top of the file and then in the immediate heading line for each block of predicted values.

`!SCORE` requests ASReml write the SCORE vector and the Average Information matrix to files <basename>`.SCO` and <basename>`.AIM`. The values written are from the last iteration.

`!SCREEN` [n] [`!SMX` m] performs a 'Regression Screen', a form of all subsets regression. For d model terms in the DENSE equations, there are $2^d - 1$ possible submodels. Since for $d > 8$, $2^d - 1$ is large, the submodels explored are reduced by the parameters n and m so that only models with at least n (default 1) terms but no more than m (default 6) terms are considered. The output (see page 28) is a report to the `.asr` file with a line for every submodel showing the sums of squares, degrees of freedom and terms in the model. There is a limit of $d = 20$ model terms in the screen. ASReml will not allow interactions to be included in the screened terms. For example, to identify which three of my set of 12 covariates best explain my dependent variable given the other terms in the model, specify `!SCREEN 3 !SMX 3`. The number of models evaluated quickly increases with d but ASReml has an arbitrary limit of 900 submodels evaluated. Use the `!DENSE` qualifier to control which terms are screened. The screen is conditional on all other terms (those in the SPARSE equations) being present. A sample of Screen output is displayed at the end of this chapter.

`!TOLERANCE` [s_1 [s_2]] modifies the ability of ASReml to detect singularities in the mixed model equations. Normally (when no `!TOLERANCE` is qualifier specified), a singularity is declared if the adjusted sum of squares of a covariable is less than η or less than the uncorrected sum of squares $\times \eta$, where η is 10^{-8} in the first iteration and 10^{-10} thereafter. η is multiplied by 10^s where s is the first or second argument respectivly, so that it is more likely that an equation will be declared singular. Once a singularity is detected, the corresponding equation is dropped (forced to be zero) in subsequent iterations so that singularities are not expected after the the first iteration. If neither argument is supplied, 2 is assumed. If the second argument is omitted, it is given the same value as the first. This is intended for use in the rare equations when ASReml detects more singularities in later iterations (given those detected in the first iteration) despite the more stringent test applied after the first iteration.

If the problem of later singularities arises because of the low coefficient of variation of a covariable, it would be better to centre and rescale the covariable. If the degrees of freedom are correct in the first iteration, the problem will be with the variance parameters and a different variance model (or variance constraints) is required.

`!SLNFORM` [*i*] modifies the format of the `.sln` file.
`!SLNFORM -1` prevents the `.sln` file from being written.
`!SLNFORM` (or `!SLNFORM 1`) replaces multiple spaces with TAB and changes the file extension to `_sln.txt`. This makes it easier to load the solutions into Excel.
`!SLNFORM 2` replaces multiple spaces with COMMA and changes the file extension to `_sln.csv`. However, since factor labels sometimes contain COMMAS, this form is not so convenient.
`!SLNFORM 3` replaces multiple spaces with Ampersand, appends a trailing double backslash and changes the file extension to `_sln.tex` (Latex style).
Additional significant digits are reported with these other formats. Omitting the qualifier means the standard fixed field format is used.

`!SPATIAL` increases the amount of information reported on the residuals obtained from the analysis of a two dimensional regular grid field trial. The information is written to the `.res` file.

`!SUBSET` <label> <factor> <list>
forms a new factor (<label>) derived from an existing factor (<factor>) by selecting a subset (<list>) of its levels. The qualifier occupies its own line after the data filename line but before the linear model. e.g.

```
!SUBSET EnvC Env 3 5 8 9 :15 21 33
```

defines a reduced form of the factor Env just selecting the environments listed. It might then be used in the model in an interaction. The intention is to simplify the model specification in MET (Multi Environment Trials) analyses where say Column effects are to be fitted to a subset of environments. Missing values are transmitted as missing and records whose level is zero are transmitted as zero.

`!SUM` causes ASReml to report a general description of the distribution of the data variables and factors and simple correlations among the variables for those records included in the analysis. This summary will ignore data records for which the variable being analysed is missing unless a multivariate analysis is requested or missing values are being estimated. The information is written to the `.ass` file.

`!PVSFORM` [f] controls form of the `.pvs` file
`!PVSFORM 1` means TAB separated: `.pvs` becomes `_pvs.txt`
`!PVSFORM 2` means COMMA separated: .pvs becomes `_pvs.csv`
`!PVSFORM 3` means Ampersand separated: `.pvs` becomes `_pvs.tex`

`!TABFORM` [f] controls form of the `.tab` file
`!TABFORM 1` means TAB separated: `.tab` becomes `_tab.txt`
`!TABFORM 2` means COMMA separated: `.tab` becomes `_tab.csv`
`!TABFORM 3` means Ampersand separated: `.tab` becomes `_tab.tex`

`!TXTFORM` [i] sets the default argument for `!PVSFORM`, `!SLNFORM`, `!TABFORM` and `!YHTFORM` if these are not explicitly set. `!TXTFORM` [i] modifies the format of the file as follows:

`!TXTFORM` (or `!TXTFORM 1`) replaces multiple spaces with TAB and changes the file extension to `_xxx.txt`. This makes it easier to load the solutions into Excel.

`!TXTFORM 2` replaces multiple spaces with COMMA and changes the file extension to `_xxx.csv`. However, since factor labels sometimes contain COMMAS, this form is not so convenient.

`!TXTFORM 3` replaces multiple spaces with Ampersand, appends a trailing double backslash and changes the file extension to `_xxx.tex` (Latex style).
Additional significant digits are reported with these other formats. Omitting the qualifier means the standard fixed field format is used. For `.yht` and `.sln` files, setting i to -1 means the file is not formed.

`!TWOWAY` modifies the appearance of the variogram calculated from the residuals obtained when the sampling coordinates of the spatial process are defined on a lattice. The default form is based on absolute 'distance' in each direction. This form distinguishes same sign and different sign distances and plots the variances separately as two layers in the same figure.

`!VRB` requests writing of `.vrb` file. Previously, the default was to write

`!VGSECTORS` [s] The sample variogram reported by ASReml now has two forms depending on whether the spatial coordinates represent a complete rectangular lattice (as typical of a field trial) or not. In the lattice case, the sample variogram is calculated from the triple $(l_{ij1}, l_{ij2}, v_{ij})$ where $l_{ij1} = s_{i1} - s_{j1}$ and $l_{ij2} = s_{i2} - s_{j2}$ are the displacements. As there will be many v_{ij} with the same displacements, ASReml calculates the means for each displacement pair $l_{ij1}, |l_{ij2}|$ (`!TWOWAY`) or $|l_{ij1}|, |l_{ij2}|$ (default) and displays the result as a perspective plot indexed by displacement. In this case, the two directions may be on different scales and the distances are usually indexed say $1 \cdots r$ and $1 \cdots c$ for an $r \times c$ lattice.

Otherwise ASReml forms a variogram based on radial coordinates. It calculates the distance between points $d_{ij} = \sqrt{l_{ij1}^2 + l_{ij2}^2}$ and angle $\theta_{ij} = \tan^{-1}(l_{ij1}/l_{ij2})$ and averages the v_{ij} within 12 distance classes and 4, 6 or 8 (`!VGSECTORS`) direction classes and reports the trends in $\bar{v}_{ij}$ with increasing distance for each sector. `!VGSECTORS` $[s]$ requests that the variogram formed with radial coordinates be based on s (4, 6 or 8) sectors of size $180/s$ degrees. The default is 4 sectors if `!VGSECTORS` is omitted and 6 sectors if it is specified without an argument. The first sector is centred on the X direction.

ASReml also computes the variogram from random effects which appear to have a variance structures defined in terms of distance. The variogram details are reported in the `.res` file.

Figure 8.1 is the variogram using radial coordinates obtained using predictors of random effects fitted as `fac(xsca,ysca)`. It shows low semivariance in `xsca` direction, high semivariance in the `ysca` direction with intermediate values in the 45 and 135 degrees directions.

`!WMF` sets hardcopy graphics file type to `.wmf`.

`!YHTFORM` $[f]$ controls the form of the `.yht` file
 `!YHTFORM` -1 suppresses formation of the `.yht` file
 `!YHTFORM` 1 means TAB separated: `.yht` becomes `_yht.txt`
 `!YHTFORM 2` means COMMA separated: `.yht` becomes `_yht.csv`
 `!YHTFORM 3` means Ampersand separated `.yht` becomes `_yht.tex`

Finally we display a portion of Regression Screen (see `!SCREEN`) output. The qualifier was `!SCREEN 3 !SMX 3`.

```
Source                  Model  terms     Gamma     Component     Comp/SE   % C
idsize                     92     92  0.581102      0.136683        3.31   0 P
expt.idsize               828    828  0.121231      0.285153E-01    1.12   0 P
Variance                  504    438   1.00000      0.235214       12.70   0 P

Analysis of Variance                NumDF     DenDF_con F_add     F_con M P_con
113 mu                                  1       72.4 65452.25       NA   .    NA
```

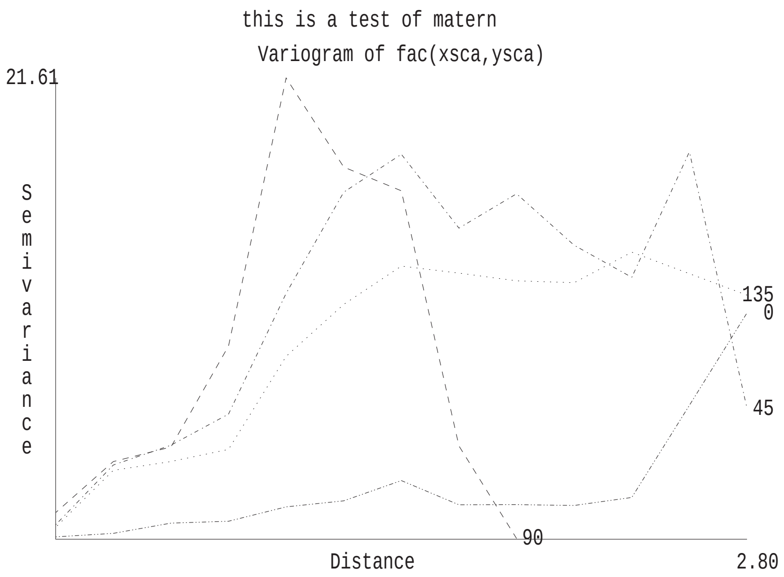

Figure 8.1 Variogram in 4 sectors for Cashmore data

```
  2 expt                               6      37.5      5.27      0.64 A 0.695

  4 type                               4      63.8     22.95      3.01 A 0.024
114 expt.type                         10      79.3      1.31      0.93 B 0.508
 23 x20                                1      55.1      4.33      2.37 B 0.130
 24 x21                                1      63.3      1.91      0.87 B 0.355
 25 x23                                1      68.3     23.93      0.11 B 0.745

 26 x39                                1      79.7      1.85      0.35 B 0.556
 27 x48                                1      69.9      1.58      2.08 B 0.154
 28 x59                                1      49.7      1.41      0.08 B 0.779
 29 x60                                1      59.6      1.46      0.42 B 0.518
 30 x61                                1      64.0      1.11      0.04 B 0.838

 31 x62                                1      61.8      2.18      0.09 B 0.770
 32 x64                                1      55.6     31.48      4.50 B 0.038
 33 x65                                1      57.8      4.72      6.12 B 0.016
 34 x66                                1      58.5      1.13      0.03 B 0.872
 35 x70                                1      59.3      1.71      1.40 B 0.242

 36 x71                                1      64.4      0.08      0.01 B 0.929
 37 x73                                1      59.0      1.79      3.01 B 0.088
 38 x75                                1      59.9      0.04      0.26 B 0.613
 39 x91                                1      63.8      1.44      1.44 B 0.234
Notice: The DenDF values are calculated ignoring fixed/boundary/singular
            variance parameters using empirical derivatives.

129 mv_estimates                        9 effects fitted
  9 idsize                             92 effects fitted (       7 are zero)
115 expt.idsize                       828 effects fitted (     672 are zero)
127 at(expt,6).type.idsize.meth         9 effects fitted (+    2199 singular)
```

```
 128 at(expt,7).type.idsize.meth          10 effects fitted (+    2198 singular)

LINE REGRESSION      RESIDUAL             ADJUSTED  FACTORS INCLUDED
 NO DF SUMSQUARES    DF MEANSQU R-SQUARED R-SQUARED 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23
  1  3  0.1113D+02  452  0.2460  0.09098   0.08495   1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                                   *****     *****
  2  3  0.1180D+02  452  0.2445  0.09648   0.09049   1  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0
                                   *****     *****
  3  3  0.1843D+01  452  0.2666  0.01507   0.00853   0  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0
  4  3  0.1095D+02  452  0.2464  0.08957   0.08353   1  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
  5  3  0.1271D+02  452  0.2425  0.10390   0.09795   1  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0
                                   *****     *****
  6  3  0.9291D+01  452  0.2501  0.07594   0.06981   0  1  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0
  7  3  0.9362D+01  452  0.2499  0.07652   0.07039   0  0  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0
  8  3  0.1357D+02  452  0.2406  0.11091   0.10501   1  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0
                                   *****     *****
  9  3  0.9404D+01  452  0.2498  0.07687   0.07074   0  1  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0
 10  3  0.1266D+02  452  0.2426  0.10350   0.09755   1  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
 11  3  0.1261D+02  452  0.2427  0.10313   0.09717   1  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0
 12  3  0.9672D+01  452  0.2492  0.07906   0.07295   0  1  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0
 13  3  0.9579D+01  452  0.2494  0.07830   0.07218   0  0  1  0  1  1  0  0  0  0  0  0  0  0  0  0  0
 14  3  0.9540D+01  452  0.2495  0.07797   0.07185   0  0  0  1  1  1  0  0  0  0  0  0  0  0  0  0  0
 15  3  0.1089D+02  452  0.2465  0.08907   0.08302   1  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0
 16  3  0.2917D+01  452  0.2642  0.02384   0.01736   0  1  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0
 17  3  0.2248D+01  452  0.2657  0.01838   0.01187   0  0  1  1  0  1  0  0  0  0  0  0  0  0  0  0  0
 18  3  0.1111D+02  452  0.2460  0.09088   0.08484   1  0  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0
 19  3  0.1746D+01  452  0.2668  0.01427   0.00773   0  1  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0
 20  3  0.1030D+02  452  0.2478  0.08423   0.07815   1  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
 21  3  0.1279D+02  452  0.2423  0.10454   0.09860   1  0  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0
 22  3  0.8086D+01  452  0.2527  0.06609   0.05989   0  1  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0
 23  3  0.7437D+01  452  0.2542  0.06079   0.05456   0  0  1  0  0  1  1  0  0  0  0  0  0  0  0  0  0
 24  3  0.1071D+02  452  0.2469  0.08755   0.08149   0  0  0  1  0  1  1  0  0  0  0  0  0  0  0  0  0
 25  3  0.1370D+02  452  0.2403  0.11200   0.10611   0  0  0  0  1  1  1  0  0  0  0  0  0  0  0  0  0
                                   *****     *****
 26  3  0.1511D+02  452  0.2372  0.12351   0.11770   1  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0
                                   *****     *****
 27  3  0.1353D+02  452  0.2407  0.11064   0.10473   0  1  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0
...
680  3  0.1057D+02  452  0.2472  0.08641   0.08035   1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
```

9 TABULATE

TABULATE statements may now appear before the model line as this is logically, a better place for them. If a linear (mixed) model is not supplied, ASReml will generate a simple model as it does not actually read the data until it has read a linear model line. Tabulation is of the data records included in the analysis (i.e. leaving out records elimated from the analysis model because of missing values in the variate or in the design factors).

The qualifiers for optional output are: `!COUNT`, `!SD`, `!RANGE` and `!STATS`. `!STATS` is shorthand for `!COUNT !SD !RANGE` The requested statistics are reported for each cell in the table. The tabulation includes the same records as are analysed in the subsequent linear model.

The qualifier `!DECIMALS` $[d]$ $(1 \leq d \leq 7)$ requests means be reported with d decimal places. If omitted, ASReml reports 5 significant digits; if specified without an argument, 2 is assumed.

10 Linear Model Specification

10.1 Generalized Linear Models

ASReml includes facilities for fitting the family of Generalized Linear Models (GLMs, Nelder and Wedderburn, 1974, Nelder and McCullagh, 1994). GLMs are specified by qualifiers after the name of the dependent variable but before the $\sim$ character. Table 10.1 lists the link function qualifiers which relate the linear predictor ($\boldsymbol{\eta}$) scale to the observation ($\boldsymbol{\mu}$ =E[$\boldsymbol{y}$]) scale. Table 10.2 lists the distributions and other qualifiers.

Table 10.1 Link qualifiers and functions

Qualifier	Link	Inverse Link	Available with
`!IDENTITY`	$\boldsymbol{\eta} = \boldsymbol{\mu}$	$\boldsymbol{\mu} = \boldsymbol{\eta}$	All
`!SQRT`	$\boldsymbol{\eta} = \sqrt{\boldsymbol{\mu}}$	$\boldsymbol{\mu} = \boldsymbol{\eta}^2$	Poisson
`!LOGARITHM`	$\boldsymbol{\eta} = \ln(\boldsymbol{\mu})$	$\boldsymbol{\mu} = \exp(\boldsymbol{\eta})$	Normal, Poisson, Negative Binomial, Gamma
`!INVERSE`	$\boldsymbol{\eta} = 1/\boldsymbol{\mu}$	$\boldsymbol{\mu} = 1/\boldsymbol{\eta}$	Normal, Gamma, Negative Binomial
`!LOGIT`	$\boldsymbol{\eta} = \boldsymbol{\mu}/(1-\boldsymbol{\mu})$	$\boldsymbol{\mu} = \frac{1}{(1+\exp(-\boldsymbol{\eta}))}$	Binomial
`!PROBIT`	$\boldsymbol{\eta} = \Phi^{-1}(\boldsymbol{\mu})$	$\boldsymbol{\mu} = \Phi(\boldsymbol{\eta})$	Binomial
`!COMPLOGLOG`	$\boldsymbol{\eta} = \ln(-\ln(1-\boldsymbol{\mu}))$	$\boldsymbol{\mu} = 1 - e^{-e^{\boldsymbol{\eta}}}$	Binomial

where μ is the mean on the data scale and $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\tau}$ is the fitted value on the underlying scale.

A second dependent variable may be specified if a bivariate analysis is required but it will always be treated as a normal variate (no syntax is provided for specifying GLM attributes for it). The `!ASUV` qualifier is required in this situation for the GLM weights to be utilized.

Table 10.2: GLM qualifiers

qualifiers	action
Distributions	where μ is the mean on the data scale calculated from $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\tau}$,$n$ is the count specified by the `!TOTAL` qualifier, v is the variance expression for the distribution, d is the deviance expression for the distribution, y is the observation and ϕ is a parameter set with the `!PHI` qualifier. The default link is listed first followed by permitted alternatives.
`!NORMAL` [`!LOGARITHM` | `!INVERSE`]	The model is fitted on the log/inverse scale but the residuals are on the natural scale.
`!BINOMIAL` [`!LOGIT` | `!IDENTITY` | `!PROBIT` | `!COMPLOGLOG`] [`!TOTAL` n]	
$v = \mu(1-\mu)/n$ $d = 2n(y\ln(y/\mu)$ $+(1-y)\ln(\frac{1-y}{1-\mu}))$	Proportions or counts [r] are indicated if `!TOTAL` specifies the variate containing the binomial totals. Proportions are assumed if no response value exceeds 1. A binary variate [0, 1] is indicated if `!TOTAL` is unspecified. The expression for d on the left applies when y is proportions (or binary). The logit is the default link function. The variance on the underlying scale is $\pi^2/3 \sim 3.3$ (underlying logistic distribution) for the logit link.
`!POISSON` [`!LOGARITHM` | `!IDENTITY` | `!SQRT`]	
$v = \mu$ $d = 2(y\mathtt{ln}(y/\mu)$ $-(y-\mu))$	Natural logarithms are the default link function. ASReml assumes the Poisson variable is not negative.
`!GAMMA` [`!INVERSE` | `!IDENTITY` | `!LOGARITHM`] [`!PHI` ϕ] [`!TOTAL` n]	
$v = \mu^2/(\phi n)$ $d = 2n(-\phi\mathtt{ln}(\frac{\phi y}{\mu})$ $+\frac{\phi y-\mu}{\mu})$	The inverse is the default link function. n is defined with the `!TOTAL` qualifier and would be degrees of freedom in the typical application to mean-squares. The default value of ϕ is 1.
`!NEGBIN` [`!LOGARITHM` | `!IDENTITY` | `!INVERSE`] [`!PHI` ϕ]	
$v = \mu + \mu^2/\phi$ $d = 2((\phi+y)\mathtt{ln}(\frac{\mu+\phi}{y+\phi})$ $+y\mathtt{ln}(\frac{y}{\mu}))$	fits the Negative Binomial distribution. Natural logarithms are the default link function. The default value of ϕ is 1.
General qualifiers	
`!AOD` New Warn	requests an Analysis of Deviance table be generated. This is formed by fitting a series of sub models for terms in the DENSE part building up to the full model, and comparing the deviances. It is not available in association with the `PREDICT`.For example `LS !BIN !TOT COUNT !AOD ~ mu SEX GROUP`

GLM qualifiers

qualifier	action
`!DISP [`h`]`	includes an *overdispersion* scaling parameter (h) in the weights. If `!DISP` is specified with no argument, ASReml estimates it as the residual variance of the working variable. Traditionally it is estimated from the deviance residuals, reported by ASReml as `Variance heterogeneity`. For example, `count !POIS !DISP ~ mu group`
`!OFFSET [`o`]`	is used especially with binomial data to include an offset in the model where o is the number or name of a variable in the data. The offset is only included in binomial and Poisson models (for Normal models just subtract the offset variable from the response variable), for example `count !POIS !OFFSET base !disp ~ mu group` The offset is included in the model as $\boldsymbol{\eta} = \mathbf{X}\tau + o$. The offset will often be something like $\ln(n)$.
`!TOTAL [`v`]`	is used especially with binomial data where v is the field containing the total counts for each sample. If omitted, count is taken as 1.
Residual	qualifiers control the form of the residuals returned in the `.yht` file. The predicted values returned in the `.yht` file will be on the linear predictor scale if the `!WORK` or `!PVW` qualifiers are used. They will be on the observation scale if the `!DEVIANCE`, `!PEARSON`, `!RESPONSE` or `!PVR` qualifiers are used.
`!DEVIANCE`	produces deviance residuals, the signed square root of d/h from Table 10.2 where h is the dispersion parameter controlled by the `!DISP` qualifier. This is the default.
`!PEARSON`	produces Pearson residuals, $\frac{y-\mu}{\sqrt{v}}$
`!PVR`	writes fitted values on the response scale in the `.yht` file. This is the default.
`!PVW`	writes fitted values on the linear predictor scale in the `.yht` file.
`!RESPONSE`	produces simple residuals, $y - \mu$
`!WORK`	produces residuals on the linear predictor scale, $\frac{y-\mu}{d\mu/d\eta}$

10.2 Generalized Linear Mixed Models

This section was written by Damian Collins

There is the capacity to fit a wider class of models which include additional random effects for non-normal error distributions. The inclusion of random terms in a GLM is usually referred to as a Generalized Linear Mixed Model (GLMM). For GLMMs, ASReml uses what is commonly referred to as penalized quasi-likelihood or PQL (Breslow and Clayton, 1993). The technique is also known by other names, including Schall's technique (Schall, 1991), pseudo-likelihood (Wolfinger and O'Connell, 1993) and joint maximisation (Harville and Mee, 1984, Gilmour *et al.*, 1985). It is implemented in many statistical packages, for instance, in the GLMM procedure (Welham, 2005) and the IRREML procedure of Genstat (Keen, 1994), in MLwiN (Goldstein *et al.*,1998), in the GLMMIXED macro in SAS and in the GLMMPQL function in R, to name a few.

The PQL technique is based on a first order Taylor series approximation to the likelihood. It has been shown to perform poorly for certain types of GLMMs. In particular, for binary GLMMs where the number of random effects is large compared to the number of observations, it can underestimate the variance components severely (50%) (e.g. Breslow and Lin, 1995, Goldstein and Rasbash, 1996, Rodriguez and Goldman, 2001). For other types of GLMMs, such as Poisson data with many observations per random effect, it has been reported to perform quite well (e.g. Breslow, 2003). As well as the above references, users can consult McCulloch and Searle (2001) for more information about GLMMs.

Most studies investigating PQL have focussed on estimation bias. Much less attention has been given to the wider inferential issues such as hypothesis testing. In addition, the performance of this technique has only been assessed on a small set of relatively simple GLMMs. Anecdotal evidence from users suggests that this technique can give very misleading results in certain situations.

Caution

Therefore we cannot recommend the use of this technique for general use. It is included in the current version of ASReml for advanced users. It is highly recommended that its use be accompanied by some form of cross-validatory assessment for the specific dataset concerned. For instance, one way of doing this would be by simulating data using the same design and using parameter values similar to the parameter estimates achieved, such as used in Millar and Willis (1999).

Caution

The standard GLM Analysis of Deviance (`!AOD`) should not be used when there are random terms in the model as the variance components are reestimated for each submodel.

10.3 New Model terms.

`at(F,n)` is extended so that `at(F,i).X at(F,j).X at(F,k).X` can be written as `at(F,i,j,k).X` NB The `at(F,i,j,k)` term must be the first component of the interaction. Any number of levels may be listed.

Caution

`at(F,i)` is extended so that `at(F)` generates `at(F,i)` for all levels of F. NB. Since this command is interpreted before the data is read, it is necessary to declare the number of levels correctly in the field definition. This extended form may only be used as the first term in an interaction.

`ge(F,n)`, `gt(F,n)`, `le(F,n)` and `lt(F,n)` create binary covariates indicating whether the level code for factor F is greater (less) than the argument n. They are similar to `at(F,n)` which indicates whether the level code equals n.

Caution

`h(`F`)` requests ASReml to fit the model term for factor F using Helmert constraints; these are the standard default constraints used by S-plus. Neither Sum zero nor Helmert constraints generate interpretable effects if singularities occur. ASReml runs more efficiently if no constraints are applied. Following is an example of Helmert covariables for a factor with 5 levels.

	C1	C2	C3	C4
F1	-1	-1	-1	-1
F2	1	-1	-1	-1
F3	0	2	-1	-1
F4	0	0	3	-1
F5	0	0	0	4

`out(`i`)`, `out(`i`,`t`)` establishes a binary variable which is:

`out(`i`)` 1 if data relates to observation i, (trait 1), else is 0

`out(`i`,`t`)` 1 if data relates to observation i, (trait t), else is 0

The intention is that this be used to test/remove single observations for example to remove the influence of an outlier or influential point. Possible outliers will be evident in the plot of residuals vs fitted values (see the `.res` file) and the appropriate record numbers for the `out()` term are reported in the `.res` file. Note that i relates to the data analysed and will not be the same as the record number as obtained by counting data lines in the data file if there were missing observations in the data and they have not been estimated. (To drop records based on the record nuber in the data file, use the `!D` transformation in association with the `!=V0` transformation.)

`pow(`x, p`[,`o`])` defines the covariable $(x + o)^p$ for use in the model where x is a variable in the data, p is a power and o is an offset. `pow(`x,`0.5[,`o`])` is equivalent to `sqr(`x`[,`o`])`; `pow(`x,`0[,`o`])` is equivalent to `log(`x`[,`o`])`; `pow(`x,`-1[,`o`])` is equivalent to `inv(`x`[,`o`])`.

`qtl(`M`,`s`)` calculates an expected marker state from flanking marker information at position s of the linkage group M(see `!MM` to define marker locations). s should be given in Morgans.

11 PREDICT

11.1 General

The order terms appear in the predict table is now controlled by the user: they appear in the order in which the user specifies them on the `predict` directive.

The syntax is extended to allow specific levels to be specified by name as well as by position. For example

```
    PREDICT Sex 'male'
```

The `!DEC` *n* qualifier gives the user control of the number of decimal places reported in the predict table where *n* is 0...9. The default is 4. `G15.9` format is used if *n* exceeds 9. When `!VVP` or `!SED` are used, the values are displayed with 6 significant digits unless *n* is specified and even when the values are displayed with 9 significant digits.

`!TDIFF` requests *t*-statistics be printed for all combinations of predicted values.

A second `!PRESENT` qualifier is allowed on a `PREDICT` statement (but not with `!PRWTS`). This is needed when there are two nested factors such as sites within regions and genotype within family. The two lists must not overlap.

11.2 Complicated weighting

Generally, when forming a prediction table, it is necessary to average over (or ignore) some potential dimensions of the prediction table. By default, ASReml uses equal weights ($1/f$ for a factor with f levels). More complicated weighting is achieved by using the `!AVERAGE` qualifier to set specific (unequal) weights for each level of a factor. However, sometimes the weights to be used need to be defined with respect to two or more factors. The simplest case is when there are missing cells and weighting is equal for those cells in a multiway table that are present; achieved by using the !PRESENT qualifier. This is now further generalized by allowing the user to fill in the weights for use by the !PRESENT machinery via the !PRWTS qualifier.

The user specifies the size of the terms in the multiway table with the !PRESENT statement and then gives a table of weights using the !PRWTS qualifier. There may only be one !PRESENT qualifier on the predict line when !PRWTS is specified. The order of factors in the tables of weights must correspond to the order in the !PRESENT list with later factors nested within preceding factors. Check the output to ensure that the values in the tables of weights are applied in the correct order. ASReml may transpose the table of weights to match the order it needs for processing.

Caution

We consider a rather complicated example from a rotation experiment conducted over several years. This particular analysis followed the analysis of the daily live weight gain per hectare of the sheep grazing the plots. There were periods when no sheep grazed. Different flocks grazed in the different years. Daily liveweight gain was assessed between 5 and 8 times in the various years. To obtain a measure of total productivity in terms of sheep liveweight, we need to weight the daily per sheep figures by the number of sheep grazing days per month. To obtain treatment effects for each year, the experimenter used

```
predict year 1 crop 1 pasture lime !AVE month 56 55 56 53 57 63 6*0
predict year 2 crop 1 pasture lime !AVE month 36 0 0 53 23 24 54 54 43 35 0 0
predict year 3 crop 1 pasture lime !AVE month 70 0 21 17 0 0 0 70 0 0 53 0
predict year 4 crop 1 pasture lime !AVE month 53 56 22 92 19 44 0 0 36 0 0 49
predict year 5 crop 1 pasture lime !AVE month 0 22 0 53 70 22 0 51 16 51 0 0
```

but then wanted to average over years as well. Both of the following predict statements produce the required values.

```
predict crop 1 pasture lime !PRES year month !PRWTS { 56 55 56 53 57 63 0 0 0,
 0 0 0 36 0 0 53 23 24 54 54 43 35 0  0 70 0  21 17 0  0  0  70 0  0  53 0 53,
 56 22 92 19 44 0  0  36 0  0  49 0  22 0  53 70 22 0  51 16 51 0  0}/5
predict crop 1 pasture lime !PRES month year !PRWTS { 56 36 70 53 0,
 55 0 0 56 22  56 0 21 22 0  53 53 17 92 53  57 23 0 19 70  63 24 0 44 22,
 0 54 0  0 0   0 54 70 0 51   0 43  0 36 16   0 35 0  0 51  0 0 53 5*0 49 0}/5
```

We have presented both sets of predict statements to show how the weights were derived and presented. Notice that the order in !PRESENT year month implies that the weight coefficients are presented with values for months nested within years. There is a check which reports if non zero weights are associated with cells that have no data. The weights are reported in the `.pvs` file. `!PRESENT` counts are reported in the `.res` file.

`!TURNINGPOINTS` *n* requests ASReml to scan the predicted values from a fitted line for possible turning points and if found, report them and save them internally in a vector which can be accessed by subsequent parts of the same job using `!TP` *n*. This was added to facilitate location of putative QTL.

`!TWOSTAGEWEIGHTS` is intended for use with variety trials which will subsequently be combined in a meta analysis. It forms the variance matrix for the predictions, inverts it and writes the predicted variety means with the corresponding diagonal elements of this matrix to the `.pvs` file. These values are used in some variety testing programs in Australia for a subsequent second stage analysis across many trials. A data base is used to collect the results from the individual trials and write out the combined data set. The diagonal elements are used as weights in the combined analysis.

11.3 PLOT graphic control qualifiers

The `!PLOT` qualifier produces a graphic of the predictions. Where there is more than one prediction factor, a multi-panel 'trellis' arrangement may be used. Alternatively, one or more factors can be superimposed on the one panel. The data can be added to the plot to assist informal examination of model fit.

With no plot options, ASReml chooses an arrangement for plotting the predictions by recognising any covariates and noting the size of factors. However, the user is able to customize how the predictions are plotted by either using options to the `!PLOT` qualifier or by using the graphical interface. The graphical interface is accessed by typing Esc when the figure is displayed.

The `!PLOT` qualifier has the following options:

`^adddata` `^addlabels` factor-list `^addlines` factor-list
`^noSEs` `^semult` m `^joinmeans`
`^xaxis` factor `^superimpose` factor-list `^condition` factor-list
`^goto` page-number `^saveplot` filename `^layout` rows cols `^bycols`
`^charlabsize` n `^panelcharsize` n `^vertxlab` `^abbrxlab` n `^abbrslab` n

Lines and data

`^addData` - superimposes the raw data.

`^addlabels` factor-list - superimposes the raw data with the data points labelled using the given factors (which must not be prediction factors). This option may be useful to identify individual data points on the graph – for instance, potential outliers – or alternatively, to identify groups of data points (e.g. all data points in the same stratum).

`^addlines` factor-list - superimposes the raw data with the data points joined using the given factors which must not be prediction factors. This option may be useful where the data are repeated measures on the same units.

`^noSEs` specifies that no error bars should be plotted (by default, they are plotted)

`^semult` number - specifies the multiplier of the SE used for creating error bars (default=1.0)

`^joinmeans` - specifies that the predicted values should be joined by lines (by default, they are only joined if the x-axis variable is numeric)

Predictions involving two or more factors

`^xaxis` factor - specifies the prediction factor to be plotted on the x-axis

`^superimpose` factor-list - specifies the prediction factors to be superimposed on the one panel.

`^condition` factor-list - specifies the conditioning factors which define the panels. These should be listed in the order that they will be used.

The factors listed must be prediction factors. If these arguments are used, all prediction factors (except for those specified with only one prediction level) must be listed once and only once, otherwise these arguments are ignored.

Layout

`^goto` page-number - specifies the page to start at, for multi-page predictions.

`^saveplot` filename - specifies the name of the file to save the plot to.

`^layout` rows cols - specifies the panel layout on each page

`^bycols` - specifies that the panels be arranged by columns (default is by rows)

`^blankpanels` n - specifies that each page contains n blank panels. This sub-option can only be used in combination with the layout sub-option.

`^extrablanks` n and `^extraspan` p specifies that an additional n blank panels be used every p pages This sub-option can only be used in combination with the `layout` sub-option.

Improving the graphical appearance (and readability)

`^labcharsize` n - specifies the relative size of the data points/labels (default=0.4)

`^panelcharsize` n - specifies the relative size of the labels used for the panels (default=1.0)

`^vertxlab` - specifies that vertical annotation be used on the x-axis (default is horizontal).

`^abbrdlab` n - specifies that the labels used for the data be appreviated to n characters.

`^abbrxlab` n - specifies that the labels used for the x-axis annotation be appreviated to *n* characters.

`^abbrslab` n - specifies that the labels used for superimposed factors be abbreviated to n characters.

12 Variance structures

12.1 Models

Autoregressive models have been extended to include `AR3` and `SAR2`. `SAR2` is a constrained form of `AR3` which represents the situation of competition from neighbouring plots plus general spatial correlation.

`CHOL`n`C` is an alternative zeroed form of `CHOL`n defining a reduced cholesky factorization. `CHOL`n models use structure $V = LDL'$ where L has 1.0 on the diagonal and zeros above the diagonal; D is diagonal. `CHOL1` extends this by setting lower triangle elements of L to zero except the first off diagonal band.

`CHOL1C` extends this by setting lower triangle elements of L to zero except the first column of L. The `CHOL`n`C` form is somewhat similar to a Factor Analytic model.

Matérn class

The Matérn class of isotropic covariance models is now described. ASReml uses an extended Matérn class which accomodates geometric anisotropy and a choice of metrics for random fields observed in two dimensions. This extension, described in detail in Haskard (2006), is given by

$$\rho(\boldsymbol{h};\boldsymbol{\phi}) = \rho_M(d(\boldsymbol{h};\delta,\alpha,\lambda);\phi,\nu)$$

where $\boldsymbol{h} = (h_x, h_y)^T$ is the spatial separation vector, (δ, α) governs geometric anisotropy, (λ) specifies the choice of metric and (ϕ, ν) are the parameters of the Matérn correlation function. The function is

$$\rho_M(d;\phi,\nu) = \left\{2^{\nu-1}\Gamma(\nu)\right\}^{-1}\left(\frac{d}{\phi}\right)^{\nu} K_\nu\left(\frac{d}{\phi}\right), \qquad (12.1)$$

where $\phi > 0$ is a range parameter, $\nu > 0$ is a smoothness parameter, $\Gamma(\cdot)$ is the gamma function, $K_\nu(.)$ is the modified Bessel function of the third kind of order

ν (Abramowitz and Stegun, 1965, section 9.6) and d is the distance defined in terms of X and Y axes: $h_x = x_i - x_j$; $h_y = y_i - y_j$; $s_x = \cos(\alpha)h_x + \sin(\alpha)h_y$; $s_y = \cos(\alpha)h_x - \sin(\alpha)h_y$; $d = (\delta|s_x|^\lambda + |s_y|^\lambda/\delta)^{1/\lambda}$.

For a given ν, the range parameter ϕ affects the rate of decay of $\rho(\cdot)$ with increasing d. The parameter $\nu > 0$ controls the analytic smoothness of the underlying process $\boldsymbol{u}_s$, the process being $\lceil\nu\rceil - 1$ times mean-square differentiable, where $\lceil\nu\rceil$ is the smallest integer greater than or equal to ν (Stein, 1999, page 31). Larger ν correspond to smoother processes. ASReml uses numerical derivatives for ν when its current value is outside the interval [0.2,5].

When $\nu = m+\frac{1}{2}$ with m a non-negative integer, $\rho_M(\cdot)$ is the product of $\exp(-d/\phi)$ and a polynomial of degree m in d. Thus $\nu = \frac{1}{2}$ yields the exponential correlation function, $\rho_M(d;\phi,\frac{1}{2}) = \exp(-d/\phi)$, and $\nu = 1$ yields Whittle's elementary correlation function, $\rho_M(d;\phi,1) = (d/\phi)K_1(d/\phi)$ (Webster and Oliver, 2001).

When $\nu = 1.5$ then

$$\rho_M(d;\phi,1.5) = \exp(-d/\phi)(1 + d/\phi)$$

which is the correlation function of a random field which is continuous and once differentiable. This has been used recently by Kammann and Wand (2003). As $\nu \to \infty$ then $\rho_M(\cdot)$ tends to the gaussian correlation function.

The metric parameter λ is not estimated by ASReml ; it is usually set to 2 for Euclidean distance. Setting $\lambda = 1$ provides the cityblock metric, which together with $\nu = 0.5$ models a separable AR1×AR1 process. Cityblock metric may be appropriate when the dominant spatial processes are alighned with rows/columns as occurs in field experiments. Geometric anisotropy is discussed in most geostatistical books (Webster and Oliver, 2001, Diggle *et al.*, 2003) but rarely are the anisotropy angle or ratio estimated from the data. Similarly the smoothness parameter ν is often set a-priori (Kammann and Wand, 2003, Diggle *et al.*, 2003). However Stein (1999) and Haskard (2006) demonstrate that ν can be reliably estimated even for modest sized data-sets, subject to caveats regarding the sampling design.

The syntax for the Matérn class in ASReml is given by `MAT`k where k is the number of parameters to be specified; the remaining parameters take their default values. Use the `!G` qualifier to control whether a specified parameter is estimated or fixed. The order of the parameters in ASReml, with their defaults, is (ϕ, $\nu = 0.5$, $\delta = 1$, $\alpha = 0$, $\lambda = 2$). For example, if we wish to fit a Matérn model with only ϕ estimated and the other parameters set at their defaults then we use `MAT1`.

MAT2 allows ν to be estimated or fixed at some other value (for example MAT2 .2 1 !GPF). The parameters ϕ and ν are highly correlated so it may be better to manually cover a grid of ν values.

We note that there is non-uniqueness in the anisotropy parameters of this metric $d(\cdot)$ since inverting δ and adding $\frac{\pi}{2}$ to α gives the same distance. This non-uniqueness can be removed by constraining $0 \leq \alpha < \frac{\pi}{2}$ and $\delta > 0$, or by constraining $0 \leq \alpha < \pi$ and either $0 < \delta \leq 1$ or $\delta \geq 1$. With $\lambda = 2$, isotropy occurs when $\delta = 1$, and then the rotation angle α is irrelevant: correlation contours are circles, compared with ellipses in general. With $\lambda = 1$, correlation contours are diamonds.

12.2 Variance model qualifiers

!=<list> has extended syntax as follows: <list> specifies one character per parameter. 1-9 are different from a-z which are different from A-Z so that 61 equalities can be specified. 0 and . mean unconstrained. A colon generates a sequence viz. a:e is the same as abcde
Putting % as the first character in <list> makes the interpretation of codes absolute (so that they apply across structures.
Putting * as the first character in <list> indicates that numbers are repeat counts, A-Z and a-z are equality codes and only . is unconstrained. Thus !=*.3A2. is equivalent to !=0AAA00 or !=0aaa00)

13 OUTPUT

A hardcopy Line printer plot of Residuals vs Predicted values is now printed to the `.res` file with other residual statistics.

A variogram is now reported for non-regular spatial data residuals. The variogram is computed for residuals when positions are indexed by two variables, and for fac(x,y) factors which have a spatial variance structure fitted. It is written to file with name tag _V.

`!PRINT` output to `.asp` file is better formatted.

13.1 Timing a job

Overall timing of a job can be obtained by comparing the start and finish times reported in the `.asr` file. It is possible to break the timing down into particular tasks by using the command line options `DL` to produce a `.asl` file containing, among other diagnostic information, times for operational components. Use the Unix system command `grep ">>>"` to extract the timing information.

14 Analysis of Variance procedures

14.1 Introduction

Inference for fixed effects in linear mixed models introduces some difficulties. In general, the methods used to construct F-tests in analysis of variance and regression cannot be used for the diversity of applications of the general linear mixed model available in ASReml. One approach would be to use likelihood ratio methods (see Welham and Thompson, 1997) although their approach is not easily implemented.

Wald-type test procedures are generally favoured for conducting tests concerning $\boldsymbol{\tau}$. The traditional Wald statistic to test the hypothesis $H_0: \boldsymbol{L\tau} = \boldsymbol{l}$ for given $\boldsymbol{L}$, $r \times p$, and $\boldsymbol{l}$, $r \times 1$, is given by

$$\mathcal{W} = (\boldsymbol{L}\hat{\boldsymbol{\tau}} - \boldsymbol{l})'\{\boldsymbol{L}(\boldsymbol{X}'\boldsymbol{H}^{-1}\boldsymbol{X})^{-1}\boldsymbol{L}'\}^{-1}(\boldsymbol{L}\hat{\boldsymbol{\tau}} - \boldsymbol{l}) \tag{14.1}$$

and asymptotically, this statistic has a chi-square distribution on r degrees of freedom. These are marginal tests, so that there is an adjustment for all other terms in the fixed part of the model. It is also anti-conservative if p-values are constructed because it assumes the variance parameters are known.

The small sample behaviour of such statistics has been considered by Kenward and Roger (1997) in some detail. They presented a scaled Wald statistic, together with an F-approximation to its sampling distribution which they showed performed well in a range (though limited in terms of the range of variance models available in ASReml) of settings.

In the following we describe the facilities now available in ASReml for conducting inference concerning terms which are the in dense fixed effects model component of the general linear mixed model. These facilities are not available for any terms in the sparse model. These include facilities for computing two types of Wald statistics and partial implementation of the Kenward and Roger adjustments.

14.2 Incremental and Conditional Wald Statistics

The basic tool for inference is the Wald statistic defined in equation 14.1. However, there are several ways $\boldsymbol{L}$ can be defined to construct a test for a particular model term, two of which are available in ASReml. ASReml obtains an F-statistic by dividing the Wald statistic by r, the numerator degrees of freedom. In this form it is possible to perform an approximate F test if we can deduce the denominator degrees of freedom. For balanced designs, these Wald F statistics are numerically identical to the F-tests obtained from the standard analysis of variance.

The first method for computing Wald statistics (for each term) is the so-called "incremental" form. For this method, Wald statistics are computed from an incremental sum of squares in the spirit of the approach used in classical regression analysis (see Searle, 1971). For example if we consider a very simple model with terms relating to the main effects of two qualitative factors A and B, given symbolically by

$$\mathsf{y} \sim \mathsf{1} + \mathsf{A} + \mathsf{B}$$

where the 1 represents the constant term (μ), then the incremental sums of squares for this model can be written as the sequence

$$\begin{aligned} &R(\mathsf{1}) \\ R(\mathsf{A}|\mathsf{1}) &= R(\mathsf{1},\mathsf{A}) - R(\mathsf{1}) \\ R(\mathsf{B}|\mathsf{1},\mathsf{A}) &= R(\mathsf{1},\mathsf{A},\mathsf{B}) - R(\mathsf{1},\mathsf{A}) \end{aligned}$$

where the $R(\cdot)$ operator denotes the reduction in the total sums of squares due to a model containing its argument and $R(\cdot|\cdot)$ denotes the difference between the reduction in the sums of squares for any pair of (nested) models. Thus $R(\mathsf{B}|\mathsf{1},\mathsf{A})$ represents the difference between the reduction in sums of squares between the so-called maximal "model"

$$\mathsf{y} \sim \mathsf{1} + \mathsf{A} + \mathsf{B}$$

and

$$\mathsf{y} \sim \mathsf{1} + \mathsf{A}$$

Implicit in these calculations is that

- we only compute Wald statistics for *estimable* functions (Searle, 1971, page 408),
- all variance parameters are held fixed at the current REML estimates from the maximal model

In this example, it is clear that the incremental Wald statistics may not produce the *desired* test for the main effect of A, as in many cases we would like to produce a Wald statistic for A based on

$$R(\mathsf{A}|1,\mathsf{B}) = R(1,\mathsf{A},\mathsf{B}) - R(1,\mathsf{B})$$

The issue is further complicated when we invoke “marginality” considerations. The issue of marginality between terms in a linear (mixed) model has been discussed in much detail by Nelder (1977). In this paper Nelder defines marginality for terms in a factorial linear model with qualitative factors, but later Nelder (1994) extended this concept to functional marginality for terms involving quantitative covariates and for mixed terms which involve an interaction between quantitative covariates and qualitative factors. Referring to our simple illustrative example above, with a full factorial linear model given symbolically by

$$\mathsf{y} \sim 1 + \mathsf{A} + \mathsf{B} + \mathsf{A.B}$$

then A and B are said to be marginal to A.B, and 1 is marginal to A and B. In a three way factorial model given by

$$\mathsf{y} \sim 1 + \mathsf{A} + \mathsf{B} + \mathsf{C} + \mathsf{A.B} + \mathsf{A.C} + \mathsf{B.C} + \mathsf{A.B.C}$$

the terms A, B, C, A.B, A.C and B.C are marginal to A.B.C. Nelder (1977, 1994) argues that meaningful and interesting tests for terms in such models can only be conducted for those tests which respect marginality relations. This philosophy underpins the following description of the second Wald statistic available in ASReml, the so-called “conditional” Wald statistic. This method is invoked by placing `!FCON` on the datafile line. ASReml attempts to construct conditional Wald statistics for each term in the fixed dense linear model so that marginality relations are respected. As a simple example, for the three way factorial model the conditional Wald statistics would be computed as

Term	Sums of Squares		M code
1	R(1)		.
A	R(A | 1,B,C,B.C)	= R(1,A,B,C,B.C) - R(1,B,C,B.C)	A
B	R(B | 1,A,C,A.C)	= R(1,A,B,C,A.C) - R(1,A,C,A.C)	A
C	R(C | 1,A,B,A.B)	= R(1,A,B,C,A.B) - R(1,A,B,A.B)	A
A.B	R(A.B | 1,A,B,C,A.C,B.C)	= R(1,A,B,C,A.B,A.C,B.C) - R(1,A,B,C,A.C,B.C)	B
A.C	R(A.C | 1,A,B,C,A.B,B.C)	= R(1,A,B,C,A.B,A.C,B.C) - R(1,A,B,C,A.B,B.C)	B
B.C	R(B.C | 1,A,B,C,A.B,A.C)	= R(1,A,B,C,A.B,A.C,B.C) - R(1,A,B,C,A.B,A.C)	B
A.B.C	R(A.B.C | 1,A,B,C,A.B,A.C,B.C)	= R(1,A,B,C,A.B,A.C,B.C,A.B.C) - R(1,A,B,C,A.B,A.C,B.C)	C

Of these the conditional Wald statistic for the 1, B.C and A.B.C terms would be the same as the incremental Wald statistics produced using the linear model

$$\mathsf{y} \sim 1 + \mathsf{A} + \mathsf{B} + \mathsf{C} + \mathsf{A.B} + \mathsf{A.C} + \mathsf{B.C} + \mathsf{A.B.C}$$

The preceeding table includes a so-called M (marginality) code reported by ASReml when conditional Wald statistics are presented. All terms with the highest M code letter are tested conditionally on all other terms in the model, i.e. by dropping the term from the maximum model. All terms with the preceding M code letter, are marginal to at least one term in a higher group, and so forth. For example, in the table, model term A.B has M code B because it is marginal to model term A.B.C and model term A has M code A because it is marginal to A.B, A.C and A.B.C. Model term mu (M code .) is a special case in that it is marginal to factors in the model but not to covariates.

Consider now a nested model which might be represented symbolically by

$$\mathsf{y} \sim 1 + \mathsf{REGION} + \mathsf{REGION.SITE}$$

For this model, the incremental and conditional Wald tests will be the same. However, it is not uncommon for this model to be presented to ASReml as

$$\mathsf{y} \sim 1 + \mathsf{REGION} + \mathsf{SITE}$$

with SITE identified across REGION rather than within REGION. Then the nested structure is hidden but ASReml will still detect the structure and produce a valid conditional Wald F-statistic. This situation will be flagged in the M code field by changing the letter to lower case. Thus, in the nested model, the three M codes would be ., A and B because REGION.SITE is obviously an interaction dependent on REGION. In the second model, REGION and SITE appear to be independent factors so the initial M codes are ., A and A. However they are not independent because REGION removes additional degrees of freedom from SITE, so the M codes are changed from ., A and A to ., a and A.

We strongly recommend, if you are in any doubt about the “conditional maximal” model for the conditional Wald F-statistic, that you consult the .aov file which spells out the “conditional maximal” model for each each term. We also advise users that the aim of the conditional Wald statistic is to facilitate inference for fixed effects. It is not meant to be prescriptive nor is it foolproof for every setting.

The Wald statistics are collectively presented in a summary table in the .asr file. The basic table includes the numerator degrees of freedom (denoted ν_{1i}) and the incremental Wald F-statistic for each term. To this is added the conditional Wald F-statistic and the M code if !FCON is specified. A conditional F-statistic is not reported for mu in the .asr but is in the .aov file (adjusted for covariates).

14.3 Kenward and Roger Adjustments

In moderately sized analyses, ASReml will also include the denominator degrees of freedom (`DenDF`, denoted by ν_{2i}, Kenward and Roger, 1997) and a probablity value if these can be computed. They will be for the conditional Wald F-statistic if it is reported. The `!DDF` i (see page 22) qualifier can be used to suppress the `DenDF` calculation (`!DDF -1`) or request a particular algorithmic method: `DDF 1` for numerical derivatives, `!DDF 2` for algebraic derivatives. The value in the probability column (either "P_add" or "P_con") is computed from an $F_{\nu_{1i},\nu_{2i}}$ reference distribution. When the `DenDF` is not available, it is possible, though anti-conservative to use the residual degrees of freedom for the denominator.

Kenward and Roger (1997) pursued the concept of construction of Wald-type test statistics through an adjusted variance matrix of $\hat{\boldsymbol{\tau}}$. They argued that it is useful to consider an improved estimator of the variance matrix of $\hat{\boldsymbol{\tau}}$ which has less bias and accounts for the variability in estimation of the variance parameters. There are two reasons for this. Firstly, the small sample distribution of Wald tests is simplified when the adjusted variance matrix is used. Secondly, if measures of precision are required for $\hat{\boldsymbol{\tau}}$ or effects therein, those obtained from the adjusted variance matrix will generally be preferred. Unfortunately the Wald statistics are currently computed using an unadjusted variance matrix.

14.4 Approximate stratum variances

ASReml reports approximate stratum variances and degrees of freedom for simple variance components models. For the linear mixed-effects model with variance components (setting $\sigma^2_H = 1$) where $\boldsymbol{G} = \oplus_{j=1}^{q} \gamma_j \boldsymbol{I}_{b_j}$, it is often possible to consider a natural ordering of the variance component parameters including σ^2. Based on an idea due to Thompson (1980), ASReml computes approximate stratum degrees of freedom and stratum variances by a modified Cholesky diagonalisation of the expected (or average) information matrix. That is, if $\boldsymbol{F}$ is the average information matrix for $\boldsymbol{\sigma}$, let $\boldsymbol{U}$ be an upper triangular matrix such that $\boldsymbol{F} = \boldsymbol{U}'\boldsymbol{U}$. Further we define

$$\boldsymbol{U}_c = \boldsymbol{D}_c \boldsymbol{U}$$

where $\boldsymbol{D}_c$ is a diagonal matrix whose elements are given by the inverse elements of the last column of $\boldsymbol{U}$ ie $d_{cii} = 1/u_{ir}, i = 1, \ldots, r$. The matrix $\boldsymbol{U}_c$ is therefore upper triangular with the elements in the last column equal to one. If the vector $\boldsymbol{\sigma}$ is ordered in the "natural" way, with σ^2 being the last element, then we can define the vector of so called "pseudo" stratum variance components by

$$\boldsymbol{\xi} = \boldsymbol{U}_c \boldsymbol{\sigma}$$

Thence

$$\mathrm{Var}(\boldsymbol{\xi}) = \boldsymbol{D}_c^2$$

The diagonal elements can be manipulated to produce effective stratum degrees of freedom Thompson (1980) viz

$$\nu_i = 2\xi_i^2/d_{cii}^2$$

In this way the closeness to an orthogonal block structure can be assessed.

Bibliography

Abramowitz, M. and Stegun, I. A. (eds) (1965). *Handbook of Mathematical Functions*, Dover Publications, New York.

Breslow, N. (2003). Whither PQL?, *Technical Report 192*, UW Biostatistics Working Paper Series, University of Washington.

Breslow, N. and Lin, X. (1995). Bias correction in generalised linear mixed models with a single component of dispersion, *Biometrika* **82**: 81–91.

Breslow, N. E. and Clayton, D. G. (1993). Approximate inference in generalized linear mixed models, *Journal of the American Statistical Association* **88**: 9–25.

Browne, W. and Draper, D. (2004). A comparison of bayesian and likelihood-based methods for fitting multilevel models, *Research Report 04-01*, Nottingham Statistics Research Report 04-01.

Callens, M. and Croux, C. (2005). Performance of likelihood-based estimation methods for multilevel binary regression models, *Technical report*, Dept. of Applied Economics, Katholieke Universiteit Leuven.

Cullis, B. R., Gogel, B. J., Verbyla, A. P. and Thompson, R. (1998). Spatial analysis of multi-environment early generation trials, *Biometrics* **54**: 1–18.

Diggle, P. J., Ribeiro, P. J. J. and Christensen, O. F. (2003). An introduction to model-based geostatistics, *in* J. Moller (ed.), *Spatial Statistics and Computational Methods*, Springer-Verlag, pp. 43–86.

Engel, B. (1998). A simple illustration of the failure of PQL, IRREML and APHL as approximate ml methods for mixed models for binary data, *Biometrical Journal* **2**: 141–154.

Engel, B. and Buist, W. (1998). Bias reduction of approximate maximum likelihood estimates for heritability in threshold models, *Biometrics* **54**: 1155–1164.

Engel, B. and Keen, A. (1994). A simple approach for the analysis of generalized linear mixed models, *Statistica Neerlandica* **48**(1): 1–22.

Gilmour, A. R., Anderson, R. D. and Rae, A. L. (1985). The analysis of binomial data by a generalised linear mixed model, *Biometrika* **72**: 593–599.

Goldstein, H. and Rasbash, J. (1996). Improved approximations for multilevel models with binary response, *Journal of the Royal Statistical Society A – General* **159**: 505–513.

Goldstein, H., Rasbash, J., Plewis, I., Draper, D., Browne, W., Yang, M., Woodhouse, G. and Healy, M. (1998). *A user's guide to MLwiN*, Institute of Education, London.

Harvey, W. R. (1977). *Users' guide to LSML76*, The Ohio State University, Columbus.

Harville, D. and Mee, R. (1984). A mixed model procedure for analysing ordered categorical data, *Biometrics* **40**: 393–408.

Haskard, K. A. (2006). *Anisotropic Matérn correlation and other issues in model-based geostatistics*, PhD thesis, BiometricsSA, University of Adelaide.

Kammann, E. E. and Wand, M. P. (2003). Geoadditive models, *Applied Statistics* **52**(1): 1–18.

Keen, A. (1994). Procedure IRREML, *GLW-DLO Procedure Library Manual*, Agricultural Mathematics Group, Wageningen, The Netherlands, pp. Report LWA–94–16.

McCullagh, P. and Nelder, J. A. (1994). *Generalized Linear Models*, 2 edn, Chapman and Hall, London.

McCulloch, C. and Searle, S. R. (2001). *Generalized, Linear, and Mixed Models*, Wiley.

Millar, R. and Willis, T. (1999). Estimating the relative density of snapper in and around a marine reserve using a log-linear mixed-effects model, *Australian and New Zealand Journal of Statistics* **41**: 383–394.

Nelder, J. A. and Wedderburn, R. W. M. (1972). Generalised linear models, *Journal of the Royal Statistical Society, Series A* **135**: 370–384.

Robinson, G. K. (1991). That BLUP is a good thing: The estimation of random effects, *Statistical Science* **6**: 15–51.

Rodriguez, G. and Goldman, N. (2001). Improved estimation procedures for multilevel models with binary response: A case study, *Journal of the Royal Statistical Society A – General* **164**(2): 339–355.

Schall, R. (1991). Estimation in generalized linear models with random effects, *Biometrika* **78**(4): 719–27.

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

Stein, M. L. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*, Springer-Verlag, New York.

Stevens, M. M., Fox, K. M., Warren, G. N., Cullis, B. R., Coombes, N. E. and Lewin, L. G. (1999). An image analysis technique for assessing resistance in rice cultivars to root-feeding chironomid midge larvae (diptera: Chironomidae), *Field Crops Research* **66**: 25–26.

Waddington, D., Welham, S. J., Gilmour, A. R. and Thompson, R. (1994). Comparisons of some glmm estimators for a simple binomial model., *Genstat Newsletter* **30**: 13–24.

Webster, R. and Oliver, M. A. (2001). *Geostatistics for Environmental Scientists*, John Wiley and Sons, Chichester.

Welham, S. J. (1993). *Genstat 5 Procedure Library manual, R. W. Payne, G. M. Arnold and G. W. Morgan, eds*, release 2[3] edn, Numerical Algorithms Group, Oxford.

Welham, S. J. (2005). Glmm fits a generalized linear mixed model., *in* R. Payne and P. Lane (eds), *GenStat Reference Manual 3: Procedure Library PL17,*, VSN International, Hemel Hempstead, UK, pp. 260–265.

Wolfinger, R. and O'Connell, M. (1993). Generalized linear mixed models: A pseudo-likelihood approach, *Journal of Statistical Computation and Simulation* **48**: 233–243.

Index