

United States
Department of
Agriculture

**Agricultural
Research
Service**

August 1993
Rev: April 1995

A Manual for Use of MTDFREML

A Set of Programs to Obtain
Estimates of Variances and
Covariances

Draft

Boldman, K. G., L. A. Kriese, L. D. Van Vleck, C. P. Van Tassell and S. D. Kachman. 1995. A Manual for Use of MTDFREML. A Set of Programs To Obtain Estimates of Variances and Covariances [DRAFT]. U.S. Department of Agriculture, Agricultural Research Service.

The manual in eight chapters describes how to compile and run the FORTRAN programs for multiple-trait-restricted-maximum-likelihood estimation of genetic variances and covariances for animal models. Chapters also describe various models that can be analyzed, suggestions for computing strategies, the theory behind the method, the algorithms used so that modifications can be made, and technical descriptions of the search algorithm and its properties. The purpose is to provide not only step-by-step instructions but also background for successful use of the programs.

Note: This manual is a draft version of the technical documentation that will accompany the MTDFREML. It is being released with the software on an interim basis until the official documentation has been prepared as a publication of the U.S. Department of Agriculture's Agricultural Research Service.

Boldman, Kriese, Van Vleck, and Van Tassell are with the Roman L. Hruska U.S. Meat Animal Research Center, Agricultural Research Service, U.S. Department of Agriculture (A218 Animal Sciences, University of Nebraska, Lincoln, NE 68583-0908). Kachman is with the Institute of Agriculture and Natural Resources, University of Nebraska, Lincoln, (Department of Biometry, University of Nebraska, Lincoln, NE 68583-0712).

Microsoft and Microsoft FORTRAN are registered trademarks and Windows is a trademark of Microsoft Corporation. Microway is a registered trademark and NDP FORTRAN is a trademark of Microway, Inc. SPARSPAK is copyrighted 1984, University of Waterloo, Waterloo, Ontario, Canada. CMS and MVS/TSO are trademarks of International Business Machines Corporation.

Mention of trade names or commercial products in this report is solely for the purpose of providing specific information and does not imply recommendation or endorsement by the U.S. Department of Agriculture.

Revised April 1995

All programs and services of the U.S. Department of Agriculture are offered on a nondiscriminatory basis without regard to race, color, national origin, religion, sex, age, marital status, or handicap.

Contents

	Page
Introduction	iv
.....	iv
Program Installation	iv
Introduction and Flow Charts	v
Compile Summary	vii
CHAPTER ONE: User Notes for MTDFREML	1
Introduction	1
Computing Considerations	2
Using an INCLUDE statement	2
Microsoft FORTRAN Powerstation	3
Microsoft FORTRAN 5.1	3
MTDFREML Programs	5
MTDFNRM: Set Up A^{-1}	5
MTDFPREP: Set Up MME	6
MTDFRUN: Solve MME for (Co)Variance Components and Solutions	6
MTDFNRM: Computing Inverse of Numerator Relationship Matrix	7
Running MTDFNRM	8
MTDFPREP: Preparation for forming mixed model equations	9
Running MTDFPREP	11
MTDFRUN: Estimating Variance Components and Solving MME	14
Initial run or cold start of MTDFRUN	15
Running MTDFRUN	16
Continuation of previous run	20
Convergence???	22
Starting or Restarting MTDFRUN	22
CHAPTER TWO: Illustrations for MTDFREML	23
MTDFNRM	23
Variance Component Estimation	25
Single Trait Model	25
MTDFPREP	25
MTDFRUN	28
Multiple Trait Model	31
MTDFPREP	32
MTDFRUN	35
Contrasts, Sampling Variances and	
Expectations of Solutions	39
MTDFRUN	39

CHAPTER THREE: Alternative Models and Shared Experiences with MTDFREML	45
Sex-Limited Traits and Genetic-Environmental Interactions	45
Sire Models	46
Sire model	46
Sire and maternal grandsire model	46
Sire and dam model	47
Models with Many Levels of a Fixed Factor	48
Standard Statistical Models Without Relationships	49
Local vs Global Maximization	49
Standardization of Data	50
Sparseness	50
Contrasts	51
Number of Traits	51
Strategy for Estimation of Covariances with Multiple Traits	51
Insufficient Memory	52
 CHAPTER FOUR: Theoretical Considerations for MTDFREML	 53
Mixed Models	53
Henderson's Mixed Model Equations	53
MME for Variance Component Estimation	53
Derivative-Free REML	55
DFREML with Choleski Factorization	59
Hypothesis Tests and Prediction Error Variance	61
Multiple Trait MME for DFREML	62
Expectations of Solutions	62
Algebra of Expected Values	63
 CHAPTER FIVE: Computing Strategies for MTDFREML	 65
MTDFPREP	66
MTDFRUN	70
 CHAPTER SIX: Modifications to SPARSPAK to Find Constraints	 76
Cholesky Factorization	76
Solving Triangular Systems of Equations	77
Uniqueness of the Constraints	78
Example	78
Modified SPARSPAK Routines	80
 CHAPTER SEVEN: A Description of the Simplex Method for DFREML	 81
Simplex (Polytope) Method	81
Likelihood Evaluations Required	91
Robust and Relatively Free from Problems of Numerical Accuracy	92
Accommodate Constraints on the Parameter Space	93

CHAPTER EIGHT: Comparison of Simplex and Powell's Search Methods for DFREML	94
Introduction	94
REML	94
Multidimensional Direct Search Algorithms	95
Simplex Method	96
Powell's Method	96
A Numerical Comparison	100
Data	100
Analysis	100
Robustness	101
Simplex Method	102
Powell's Method	107
Efficiency	108
Simplex Method	108
Powell's Method	108
Conclusions	109
APPENDIX: FSPAK Invoice	111
Bibliography	112

Introduction

Program Installation

A batch file is included on the distribution diskette which interactively prompts the user for the compiler to be used (Microsoft Powerstation (MSP) , Microway NDP (MW) , Microsoft 5.1 (MS5.1), or "other") and the version of the program to install (SPARSPAK or FSPAK). Based on the user's selection a compressed copy of the files is expanded on the destination disk. To install the programs, first create a subdirectory for installation of the MTDFREML program files. This subdirectory should be the "active" subdirectory for that drive, i.e., when you chose that drive from DOS, that subdirectory should be the current one. Finally to install the programs change the active drive to the diskette drive containing the programs (e.g., A:) and type: install x <return>, where x is the drive (often C) you want to install the program files. No colon (:) should be included in this command. The install program currently supports drives C through H, but the batch file can be modified to include other drives if your installation requires that change. A set of commands that will install the MTDFREML programs in subdirectory NEWREML on drive C from diskette drive A is:

```
C:  
MD \NEWREML  
CD \NEWREML  
A:  
INSTALL C
```

The subdirectory can be given any valid DOS subdirectory name; NEWREML is used only as an example. If the hard drive is something other than C, or the diskette drive is not A:, make the appropriate changes to the above commands.

After installation, the subdirectory will contain two batch files that can be used for compiling the programs, setup.bat and compile.bat. These files must be modified on systems other than the MSP and MW FORTRAN compilers. The MS files are included as an example only on the MS5.1 and "other" installations. The setup.bat file will need to be run only once on most

systems. This file compiles subroutine libraries that will not change (e.g., MTDFREML utility subroutines, SPARSPAK, and FSPAK files). These files will usually not need to be changed and can be compiled once to create object files which can be linked to other programs as needed. The compile.bat file compiles the main MTDFREML programs, including MTDFNRM, MTDFPREP, and MTDFRUN (and its main subroutine MTDFLIK). The MTDFNRM file will likely be changed infrequently and may be moved to the setup.bat file if desired. The remaining programs MUST be recompiled if any changes are made to the PARAM.DAT file. The only exception to that requirement is a change to the sparse matrix storage vector length (MAXORDS, MAXNZE, or NHASH), in which case only MTDFRUN and MTDFLIK need to be recompiled.

For users who will be running on systems other than PCs, it would be easiest to install the programs to a PC temporarily and then move the files to the system where they will be run. For users without access to a PC, the files are compressed using PKZIP 2.04g, which is compatible with GZIP, a public domain program available from GNU software. The FSPOTH.EXE and SPKOTH.EXE contain the FSPAK and SPARSPAK versions, respectively, of the MTDFREML programs for the non-supported compilers.

Introduction and Flow Charts

For users who cannot wait to begin, the following two charts show the basic steps for compiling and running the set of three programs for MTDFREML. There are now two versions of the program, one based on the commercial version of SPARSPAK and one using the FSPAK subroutines developed by Misztal and Perez-Enciso. To use the SPARSPAK version of the programs a license from University of Waterloo, Ontario, Canada for the SPARSPAK subroutines. Because the license allows modification, the modified SPARSPAK subroutines will then be legal. The FSPAK version can be used without royalty for noncommercial purposes. Commercial users of the FSPAK version of the program should obtain a license for the FSPAK subroutines. A version of the commercial user invoice is included at the end of this manual. Parameter statements that are the same for programs MTDFPREP, MTDFRUN, and subroutine MTDFLIK are in an "include" file called PARAM.DAT. These statements may need to be modified for special models and

available computer memory. The first two chapters of this manual can be used to answer most questions that come up and are likely to be essential for less experienced users. This "short version" is for the standard FORTRAN 77 source code statements on the distribution disk that have been tested with a Microsoft Powerstation FORTRAN compiler on a DOS based personal computer. For any compilers other than the Microway NDP FORTRAN compiler, the file MSTIME.FOR will need to be included in the file to link with all of the programs. This file contains the system dependent timing routines.

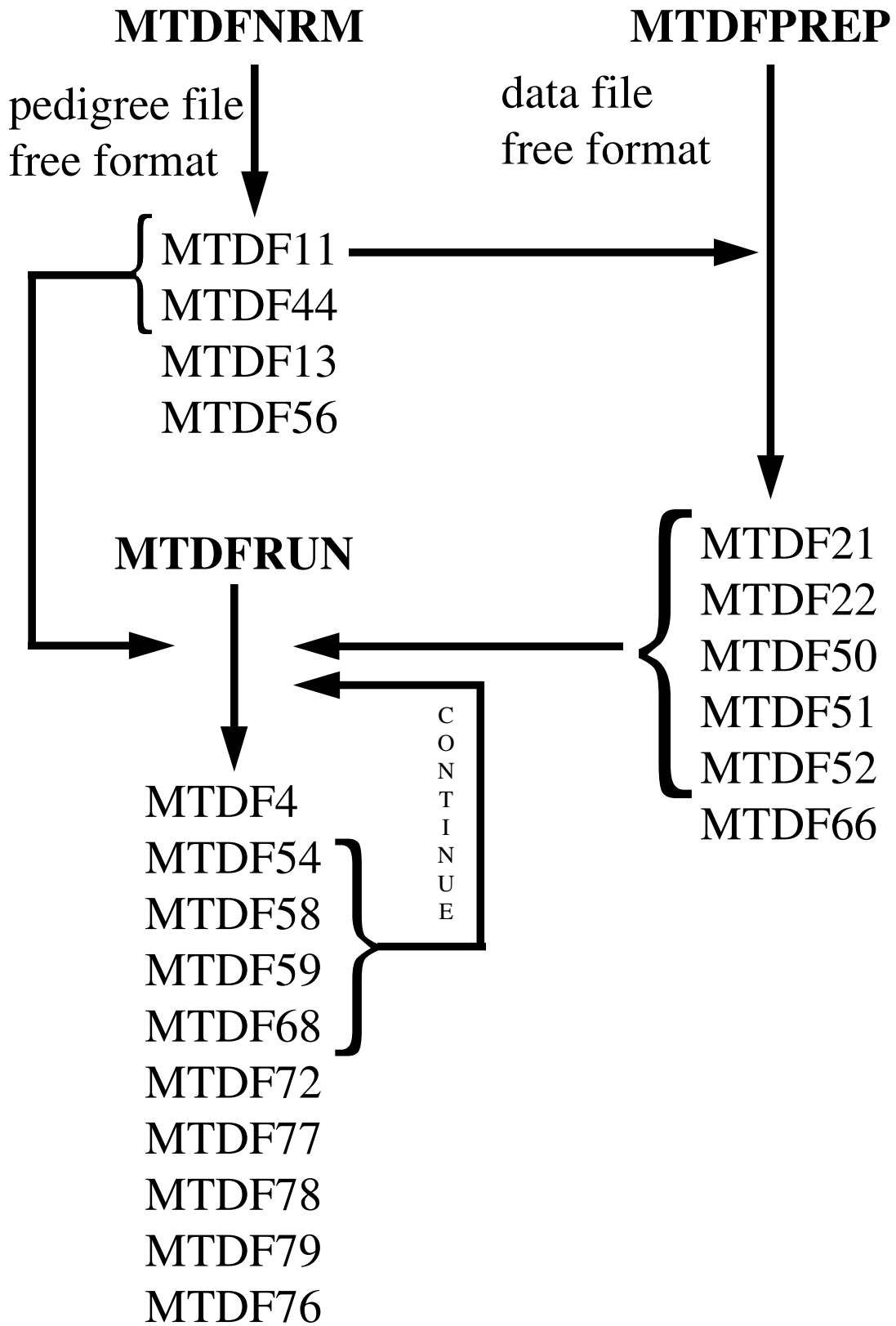
Records must be arranged with integer fixed fields to the left (identification, levels of fixed factors and other random factors) and real fields to the right (covariates and measurements of traits). Fields for missing measurements must contain a specified numerical indicator corresponding to a missing observation for that effect (e.g., 0.0, -99.99).

Good luck. If you need further help read Chapter 1, which gives step by step instructions for compiling and executing, and Chapter 2 which contains numerical examples.

Chapter 3 describes analyses for models other than animal models including non-genetic models and also provides some cautions for the use of MTDFREML and sparse matrix methods. Chapter 4 describes the algebra involved in the computations while Chapter 5 outlines the computational strategies which may help if modifications to the programs are attempted. Modifications are encouraged but the authors cannot provide much help if problems result. The difficulty of having to apply constraints to make the MME full rank as required by SPARSPAK has been overcome as shown in Chapter 6. A critical part of the MTDFREML programs is the Simplex algorithm to find the set of parameter estimates to maximize the likelihood function given the data. The Simplex procedure is described in detail in Chapter 7. Another non-linear maximization algorithm is known as Powell's method. The Simplex method and Powell's method are compared in Chapter 8 and some suggestions for use of the Simplex method are made.

Compile Summary

- MTDFNRM** Main program to compute A^{-1} and recode identification numbers of animal, sire, and dam (or sire, sire of sire, and maternal grandsire of sire). Key parameter to change is MAXAN, maximum number of animals.
Subroutines to be linked are in MTDFSUB.
- MTDFPREP** Takes data from history file of animal, recodes identification of animal (or sire) based on MTDFNRM, recodes levels of fixed and random factors to create new data file for MTDFRUN.
Include 'PARAM.DAT', a file with key parameters for vector dimensions.
Subroutines to be linked are in MTDFSUB.
- MTDFRUN** Uses data prepared by MTDFPREP to search for variance-covariance matrices to maximize the log likelihood with a DFREML algorithm.
Include 'PARAM.DAT', a file with key parameters for vectors. Subroutines to be linked are:
- MTDFSUB** Utility subroutines
- MTDFLIK** Subroutine to evaluate the logarithm of the likelihood for current values of the parameters.
- SPARSPAK** Depending on the compiler and version of the programs installed (i.e., SPARSPAK or FSPAK), there are additional files containing sparse matrix subroutines. The SPARSPAK subroutines are: in two files (SPARSPAK.FOR and MMDUPD.FOR) for the MSP compiler (the compiler crashes if MMDUPD is compiled with optimization and thus is separated), in ten files for the MW compiler (SPARSN2.F, SPARSN3.F, ... SPARSN10.F, and SPARSNG.F), in eleven files for the MS5.1 compiler (same as MW except SPARSN4.FOR is split into SPARSN4A.FOR and SPARSN4B.FOR due to compiler limitations), and in one file (SPARSPAK.FOR) for other compilers.



CHAPTER ONE: User Notes for MTDFREML

Introduction

Multiple Trait Derivative-Free Restricted Maximum Likelihood, denoted as MTDFREML, is a set of programs to estimate (co)variance components using animal models and derivative-free REML. These programs can be used for single trait, bivariate, and multiple trait animal models with repeated records including traits with sex limited expression. Solutions for fixed effects, breeding values, and uncorrelated random effects, sampling variances of solutions and contrasts and expectations of solutions can also be obtained. The statistical principles are shown in Chapter 4.

Animal models can incorporate additive genetic effects not only for animals with records, but also for parents and other relatives without records included in the pedigree file. One additional correlated random effect, (e.g., maternal genetic) and several uncorrelated random effects can be used for each trait in the analysis. Fixed effects, covariates, and uncorrelated random effects are specified separately for each trait.

Two versions of these programs are distributed. The first version incorporates a sparse matrix package, SPARSPAK (George and Ng, 1984). A license from the University of Waterloo, Ontario, Canada, is required to use SPARSPAK (Manager, Software Coordination, 200 University Avenue W., Waterloo, Ontario, N2L 3G1, Canada). The second version of the programs uses the FSPAK programs developed by Miztal and Perez-Enciso which use the public domain SPARSPAK (George and Lui, 1980) programs. These programs were modified by Elzo to include the Kachman modifications to handle singularity of the MME. The FSPAK version can be used without royalty for noncommercial puposes. Commercial users of the FSPAK version of the program should obtain a license for the FSPAK subroutines. A version of the commercial user invoice is included at the end of this manual.

The size of analyses that can be run depends on the number of traits and animals in the analysis, and computer speed and memory. For the derivative-free method, convergence for (co)variance component estimation is when the global maximum of the log likelihood function is found (see

Chapter 7). The description in this chapter assumes some mixed model, statistical and computer knowledge. Basic principles for DFREML can be found in Meyer (1991) and in Chapter 4.

Computing Considerations

All programs are written in FORTRAN 77. The simplex algorithm used in the last program (MTDFRUN) is based on Meyer's original univariate DFREML program (1988a,b).

The programs were developed on a 486 microcomputer using a Microway NDP FORTRAN 386 compiler, but should run with minor modifications on any platform with a FORTRAN compiler. At least 16 MB of memory is advisable, especially for the MTDFRUN program.

Currently, all programs have interactive input/output defined to standard FORTRAN 77 units of 5 (input from keyboard) and 6 (output to screen). Two areas that may need modification on platforms other than PC's are the input/output file connections and the timing routines. Currently, the subroutine, FCONCT (Meyer, 1988b), connects most input/output files to standard FORTRAN unit numbers. FCONCT uses standard DOS file naming conventions to connect the FORTRAN unit numbers to the program. Obviously, this will not work on all systems. Subroutines CTIME and ETIME calculate time and elapsed time in format hh.mm.ss by calling DOSTIM (a Microway NDP FORTRAN compiler function). Timing routines for the Microsoft compilers are included in a file called MTIME.FOR if Microsoft or "other" compiler is requested. For other compilers, modification of the two subroutines in the file will be necessary. On platforms where timers are not available values of zero can be returned for all variables and timing output can be ignored.

Using an INCLUDE statement

The file PARAM.DAT contains maximum parameter definitions for variables and arrays in programs MTDFPREP and MTDFRUN and subroutine MTDFLIK. The INCLUDE statement brings PARAM.DAT into MTDFPREP, MTDFRUN and MTDFLIK to provide consistent parameter definitions. In the PC environment, the INCLUDE statement looks like :

```
INCLUDE 'PARAM.DAT'
```

In MVS/TSO environments, the INCLUDE statement is :

```
INCLUDE 'QCAROL.PARAM.DAT'
```

In CMS environments, a library must be created that contains the INCLUDE source code. A possible exec to create the library containing the INCLUDE source code:

```
/* An exec to use INCLUDE statement in MTDFREML */  
TRACE RESULTS  
"MAC GEN LIB1 PARAM"  
"GLOBAL MACLIB LIB1"
```

The PARAM file *must* have a filetype of COPY in the CMS environment. After the library is created on CMS, the INCLUDE statement in MTDFPREP or MTDFRUN has the form:

```
INCLUDE(PARAM)
```

Check the system documentation for variations of this code if problems arise. Alternatively, the source code in PARAM.DAT could be placed directly into MTDFPREP and MTDFRUN programs and MTDFLIK subroutine replacing the INCLUDE statement. Note that if PARAM.DAT is changed, MTDFPREP, MTDFRUN, and MTDFLIK must be recompiled, i.e., the programs must have the same parameter definitions.

If these programs are run on a mainframe system, the type of memory used to run the programs may require a program modification. Two types of memory on mainframe machines are actual and virtual. If a mainframe has virtual memory available, known as above the line memory, accessing virtual memory for MTDFRUN and possibly MTDFNRM may be required. The MTDFLIK subroutine of MTDFRUN may need memory above the line if the analysis requires large amounts of storage space for the vector **S** (required for SPARSPAK). The vector **S** holds all non-zero MME coefficients. FORTRAN running on CMS and MVS/TSO systems requires that variables needing virtual memory be placed in a named dynamic common block and uses the @PROCESS DC option.

Microsoft FORTRAN Powerstation

The programs have been migrated to this compiler and should compile and execute without modification to source code or batch files. For a description of the compiler switches used, see the Microsoft Powerstation Fortran User's Guide.

Microsoft FORTRAN 5.1

Support for this compiler is being phased out. The following information is included for

completeness and for information for those still using this compiler. The distributed programs are no longer tested with this compiler, and the following recommendations are based on older versions of MTDFREML.

With the Microsoft FORTRAN compiler (ver. 5.1), it is advisable to use Microsoft Windows (ver. 3.1). Using a version less than 3.1 may produce erratic results. The following guidelines should be followed:

1. All files with FORTRAN source code must have an extension of .FOR.
2. All programs must be run under Windows to use more than 1 MB of extended memory. All programs need > 1 MB extended memory.
3. To answer interactive questions through a file, unit 5 must be redefined to another unit. Microsoft only allows unit 5 to be input from the keyboard, e.g., change IUN5=5 to IUN5=8.
4. Make sure to link the MSTIME file for access to timing routines.

To compile and link programs:

1. MTDNRM.FOR, MTDFSUB.FOR, and MSTIME.FOR. Switches needed to compile and link are /MW /AH and /Gt. The /MW switch incorporates Microsoft Windows libraries. Switch /AH defines huge memory models and /Gt allocates data outside the default data segment for the huge memory modules.
2. MTDFPREP.FOR, MTDFSUB.FOR, and MSTIME.FOR. Switches needed to compile and link are /MW /AH and /Gt.
3. To compile and link MTDFRUN.FOR, each component must be compiled separately. SPARSN4.FOR (which is a series of subroutines) may need to be broken into 2 parts because it may be too large to compile using the Microsoft compiler.
 - a) For each component, MTDFRUN.FOR, MTDFLIK.FOR, MTDFSUB.FOR, MSTIME.FOR, SPARSNG.FOR and SPARSN2.FOR through SPARSN10.FOR, the compiler switches need to be /MW /AH /Gt /c. Switch /c indicates compile but do not link.
 - b) In compiling each component, warning messages may appear for MTDFRUN, MTDFLIK and MTDFSUB. These are okay. Error messages will appear during the compilation of the SPARSPAK modules. These are recoverable errors, so do not be alarmed. A statement that MTDFRUN and MTDFLIK are too large for post-optimization will appear.
This is okay.
 - c) A .TXT file is needed so MTDFRUN can be linked and become an executable file. This .TXT file could be RUN.TXT, but can be called any name. The .TXT file must include the following information:

names of all programs separated by a space
name of .exe file -- leave a blank line if want it called the first program name.exe
name of map file -- leave a blank line if a map file is not wanted
name of library file with options of /packd and /seg:256
path of where fl.def file is (make sure you place a hard return at the end of this line).

ex. Sample of .txt file :

```
mtdfrun mtdflik mtdfsub mstime sparsng sparsn2 sparsn3 sparsn4 sparsn5 sparsn6 sparsn7 sparsn8 sparsn9 sparsn10 sparsn4a  
mtdfrun.exe
```

```
/nod llibfew /packd /seg:256  
c:\fortran\bin\fl.def
```

If the fl.def file is not found, check if Microsoft FORTRAN was installed for DOS and Microsoft Windows. The fl.def file is part of the Microsoft Windows installation.

d) Type in link @name of txt file.txt at C:> prompt and hit return. This will create the mtdfrun.exe file.

ex. C:>link @run.txt <return>

MTDFREML Programs

The three MTDFREML programs: 1) form the inverse of the relationship matrix, 2) prepare for set up of the weighted least squares part of MME and 3) solve the MME for (co)variance components, solutions for fixed and random effects, contrasts of solutions, sampling and contrast variances of the solutions, expectations of fixed effect solutions, and prediction error variances (PEV) of animal solutions. The following programs and subroutines are needed:

MTDFNRM: Set Up A^{-1}

MTDFNRM	Forms non-zero elements of A^{-1} using an ASCII free formatted pedigree file using the rules of Quaas (1976). Program reorders animal, sire and dam identification (for animal model) or alternatively, sire, sire of sire and maternal grandsire of sire identification (for sire models). For groups models, non-zero elements of W replace elements of A^{-1} using rules of Westell (1988).
MTDFSUB	Series of FORTRAN subroutines needed in MTDFNRM, MTDFPREP and MTDFRUN. Some routines written by K. Meyer (1988b), others by K. Boldman and P. VanRaden.

MSTIME Timing routines

MTDFPREP: Set Up MME

MTDFPREP From an ASCII data file, forms MME according to model specifications supplied to the program.

MTDFSUB Series of FORTRAN subroutines needed in MTDFNRM, MTDFPREP and MTDFRUN.

MSTIME Timing routines

MTDFP5.DAT This file is optional and will contain answers to interactive questions asked in running MTDFPREP. To use this file, FORTRAN statements to open unit 5 must have comments removed in MTDFPREP in order for the program to read answers to questions from this file instead of from the keyboard. An alternative for DOS and UNIX based systems is to execute MTDFPREP using the following format:

mtdfprep.exe < mtdfp5.dat

This assumes that the executable program is called mtdfprep.exe and the input file is mtdfp5.dat (note that any legitimate filename can be substituted). It is important to remember that filenames and commands are case sensitive in UNIX!

PARAM.DAT FORTRAN code for INCLUDE statement that contains parameter statements for maximum limits for variables such as maximum number of animals and fixed effects. If the user opts not to use the INCLUDE statement, these source statements can be placed directly into source code for MTDFPREP where the INCLUDE statement is located.

MTDFRUN: Solve MME for (Co)Variance Components and Solutions

MTDFRUN From coefficients of MME formed in MTDFPREP, estimates of (co)variance components using the SIMPLEX algorithm, and solutions for covariates, fixed and random effects, sampling and contrast variances and standard errors of solutions and expectations for fixed effect solutions can be obtained.

MTDFLIK Subroutine needed by MTDFRUN.F that creates and solves mixed model

equations and calculates the log likelihood.

MTDFSUB	Series of FORTRAN subroutines needed in MTDFNRM, MTDFPREP and MTDFRUN.
MSTIME	Timing routines
PARAM.DAT	FORTTRAN code for INCLUDE statement that contains parameter statements for maximum limits for variables such as maximum number of animals and fixed effects. If the user opts not to use the INCLUDE statement, these source statements could be placed directly into source code for MTDFPREP, MTDFRUN, and MTDFLIK where the INCLUDE statements are located. Be sure the same PARAM.DAT file is used for all three locations, i.e., if PARAM.DAT is changed, recompile MTDFPREP, MTDFRUN, and MTDFLIK.
SPARSPAK	Series of SPARSPAK modules of subroutines written in FORTRAN needed to store, factor and solve MME. The file names and numbers of files will vary with the compiler used. All versions include SPARSNG, a version of SPARSPAK subroutines modified by S. D. Kachman that automatically find dependencies in MME and set constraint equations to zero in solving MME and calculating log likelihood.
MTDFR5.DAT	<p>This file is optional and will contain answers to interactive questions asked by MTDFRUN. If this file is used, MTDFRUN reads answers from this file instead of from the keyboard. To use this file, FORTRAN statements to open unit 5 must have comments removed in MTDFRUN in order for the program to read answers to questions from this file instead of from the keyboard. An alternative for DOS and UNIX based systems is to execute MTDRUN using the following format:</p> <pre>mtdfrun.exe < mtdfr5.dat</pre> <p>This assumes that the executable program is called mtdfrun.exe and the input file is mtdfr5.dat (note that any legitimate filename can be substituted). It is important to remember that filenames and commands are case sensitive in UNIX!</p>

MTDFNRM: Computing Inverse of Numerator Relationship Matrix

Pedigree information may be in a file separate from the data when animals without records need to be included in the relationship matrix. Animals can be repeated in the data file, e.g., dairy cows may have multiple lactation records in the data set - the program will ignore duplicate pedigree

information. The file is assumed to be in free format (i.e., spaces between all variables). The source code can be easily modified to accommodate formatted read statements if animal identification numbers are not separated by spaces. The pedigree file needs to include numeric fields for:

- * Animal ID, sire ID and dam ID (optionally sire ID, sire of sire ID and maternal grandsire of sire ID for sire models).
- * If both parents of an animal *without* a record are unknown, that animal does not need to be in the pedigree file as an animal because it does not provide ties.
- * If a sire ID or dam ID is missing, the missing parent ID must be coded as a 0. If Westell's rules for group effects are used, the missing parent ID is the number for its group.
- * If an animal has missing parent(s), and if the missing parent is needed to code for a maternal or paternal correlated random effect, the missing parent must be coded with a unique number other than 0. For the groups model, the number for the missing parent must be in the animal field with its parents assigned to groups.
- * The largest ID the program can accommodate is the maximum INTEGER the compiler can handle ($2^{31} - 1 = 2,147,483,647$ for most FORTRAN compilers). If IDs are larger than this or include characters, IDs must be recoded prior to running MTDFNRM.

In the parameter statement of MTDFNRM, the maximum number of animals (MAXAN) in the relationship matrix can be changed. MAXAN represents both animals with records and base animals. [This program runs more slowly than the original DFNRM program of K. Meyer. DFNRM can be used to form the A^{-1} , but it requires a binary pedigree file, MAXAN and MAXNRM (maximum number of non-zero elements in A^{-1}) must be changed in main program and all subroutines, and the ratio between MAXAN and MAXNRM must be at least 1:10 for the program to run correctly. If DFNRM is used, the maximum ID is 99,999,999. Files produced by DFNRM are DF11 (binary) and DF44 (binary). The read statement for unit IUN11 in MTDFPREP will also need to be modified].

Running MTDFNRM

1. Compile MTDFNRM. MTDFSUB (subroutines) needs to be linked.
2. Run MTDFNRM. This program will calculate A^{-1} . The program asks :
 - a) Do you want to calculate A^{-1} for animal model (0) or sire-maternal

grandsire model (1)?

- b) Maximum animal ID in pedigree file (used for data verification).
- c) Minimum animal ID in pedigree file (used for data verification - 0 will work).
- d) Name of free formatted file containing pedigree information, e.g., ANIMAL.PED.
- e) Write pedigree file with original and recoded Ids and inbreeding.
- f) Number of integer fields in pedigree file.
- g) Position of animal ID in vector of integers.
- h) Position of sire ID in vector of integers.
- i) Position of dam ID in vector of integers.
- j) Number of genetic groups (if no groups enter 0).

The number of animals in A^{-1} will appear on the screen and in file MTDF56. This number is needed later for MTDFPREP. MTDFNRM produces three output files :

MTDF11 number of animals, followed by vector of animal IDs sorted in ascending order in ASCII format; recoded ID followed by original ID.

MTDF13 number of animals, followed by one line for each animal in the pedigree in the following format:

Recoded ID			Original ID			Inbreeding Coefficient		
Animal	Sire	Dam	Animal	Sire	Dam	Animal	Sire	Dam

MTDF44 $.5 \log |A|$ followed by A^{-1} coefficients in binary format.

MTDF56 log file of information for MTDFNRM run; number of animals, inbreeding information, etc.

MTDFPREP: Preparation for forming mixed model equations

In the data file for MTDFPREP integer variables, such as animal ID, dam ID (for maternal or permanent environment effects) and numerical identities of fixed effects must be first, followed by real variables, which include covariates and trait measurements. The data file should be in free format with at least one space separating variables. Otherwise, the source code can be modified for a formatted read of unit IUN33.

Models can be different for each trait in the analysis. The number of fixed effects, covariates

or uncorrelated random effects are usually not limiting for a trait. An INCLUDE file, PARAM.DAT, contains maximums for several variables used in the programs. The limits must be large enough to accommodate the data set. If not, error messages or wrong results will be obtained. Limits that can be changed in PARAM.DAT are :

MAXTRT	maximum number of traits in the analysis
MAXINTR	maximum number of integer variables on each record
MAXR8	maximum number of real variables on each record
MAXANIM	maximum number of animals
MAXCOV	maximum number of covariates per trait
MAXNFR	maximum number of regression coefficients per trait
MAXFIX	maximum number of fixed effects per trait
MAXNFL	maximum number of levels for each fixed effect
MAXCONS	maximum number of constraints
MAXRAN	maximum number of uncorrelated random effects per trait
MAXNRL	maximum number of levels for each uncorrelated random effect
MAXINV	maximum order of submatrix to obtain inverse elements and maximum number of elements in a contrast
MAXORDS	used by SPARSPAK version only: maximum length of S (work) vector for SPARSPAK (holds non-zero MME elements), also called MAXSA in SPARSPAK
MAXNZE	used by FSPAK version only: maximum number of non-zero elements in the coefficient matrix
NHASH	used by the FSPAK version only: length of hash table used to build list of non-zero elements of coefficient matrix. This space is reused as work space by FSPAK once the linked list is built.

Fields in the data file can be used for more than one trait and can have more than one name within or across traits. For example, for weaning weight in beef cattle, when additive, maternal and permanent environmental random effects are in the model, the dam ID field can be used to indicate both maternal genetic and permanent environment effects. More than one uncorrelated random effect can be specified for each trait in the analysis. Within trait, uncorrelated random effects will, of course, be uncorrelated. However, if the same uncorrelated random factor is used across traits, a covariance can be estimated.

The MME set up by MTDFPREP have the following order :

covariate(s) trait 1

covariate(s) trait n
 fixed effect(s) trait 1
 .
 fixed effect(s) trait n
 additive genetic animal effect trait 1
 .
 additive genetic animal effect trait n
 additional correlated random effect trait 1 (e.g., maternal genetic)
 .
 additional correlated random effect trait n
 uncorrelated random effect(s) for trait 1
 .
 uncorrelated random effect(s) for trait n

The number and types of equations in the MME depend on specific models and data. Equations in the above list that do not apply to a specific analysis do not appear. All models will have additive genetic animal effects. Uncorrelated animal effects will result from a pedigree file with all sires and dams missing. Genetic variances cannot be estimated if $A = I$ for an animal model but a sire or non-genetic model can be used with $A = I$.

Running MTDFPREP

1. Compile MTDFPREP. MTDFSUB (contains subroutines) needs to be linked and PARAM.DAT must be available.
2. Run MTDFPREP. The program reads MTDF11 and asks the following questions:
 - a) Name of data file (IUN33), e.g., ANIMAL.DAT
 - b) Description of analysis (up to 6 lines, terminated with a * in column 1 after last comment line)
 - c) Number of integer variables in each line of data file
 - d) Number of real variables in each line of data file
 - e) Number of traits in the analysis

*** Questions f-y are repeated for each trait with the exceptions of q) and r)

- f) Name of trait
- g) Position for trait in list of real variables
- h) Missing value designation for trait (e.g., 0,0.0, -999.9, etc.)

*** Questions j-l will be repeated for each covariate

- i) Number of covariates
- j) Name for first covariate
- k) Position of first covariate in list of real variables
- l) Type of covariate (linear, quadratic, etc)

*** Questions n-p will be repeated for each fixed effect

- m) Number of fixed effects
- n) Name of fixed effect
- o) Position of fixed effect in list of integer variables
- p) Write levels of fixed effect to unit 66 (MTDF66): 1 yes; 0 no
- q) Position of animal ID in list of integers (same for each trait)
- r) Number of animals in A^{-1} (from MTDFNRM)

*** If number of second animal effects > 0 answer questions t) and u)

- s) Is there a second animal (e.g., maternal) effect (1 yes; 0 no)
- t) Name of second animal effect
- u) Position of second animal effect in list of integer variables

*** Questions w-y repeated for each uncorrelated random effect

- v) Number of uncorrelated random effects (e.g., PE, litter)
- w) Name of uncorrelated random effect
- x) Position of uncorrelated random effect in list of integers
- y) Write levels of uncorrelated random to unit 66 (1 yes; 0 no)

*** Question z will be asked if there is at least one covariate or fixed effect

- z) Save labels to match with mean estimates for covariates and fixed effects in MTGSRUN (1 yes; 0 no)

*** Question aa will be asked if there is at least one uncorrelated random effect

- aa) Save labels to match with mean estimates for uncorrelated random effects in MTGSRUN (1 yes; 0 no)

If the option to write levels of fixed effects or uncorrelated random effects to unit 66 (MTDF66) is 1, summary statistics for each level will be written to the output log. With many levels of a fixed effect or uncorrelated random effect, answer no (0) to avoid a large output log.

After gaining familiarity with the program, putting all the analysis information in a file (MTDFP5.DAT) for the program to read is easier than entering the data interactively. However, please enter the data interactively to become familiar with the questions the first few times. If a mistake is made answering questions interactively, the program must be started from the beginning. To change the program to read answers from an input file, remove the appropriate comment lines in MTDFPREP.F and/or MTDFRUN.F in the file definition section for unit 5 or for DOS and UNIX systems with exe files execute MTDFPREP withmtdfprep<mtdfp5.dat. See Chapter 2, Examples, for how to set up MTDFP5.DAT.

MTDFPREP produces the following files :

MTDF21 - Labels for covariate and fixed effect labels if requested

MTDF22 - Labels for uncorrelated random effects if requested

MTDF50 - information on model used in MTDFRUN. Information includes :

- * Number of traits, effects, animals, regression coefficients, equations, and columns that contain random effects.
- * Number of covariates by trait.
- * Number of regression coefficients by trait (if no. covariates > 0).
- * Number of fixed effects by trait.
- * Number of levels for each fixed effect by trait (if no. fixed effects > 0).
- * Starting equation number for direct effects by trait.
- * Number of second animal (or maternal) effects by trait.
- * Starting equation number for maternal effects by trait (if no. maternal > 0).
- * Number of uncorrelated random effects by trait
- * Number of levels for each uncorrelated random group, column of uncorrelated random group in data set (if no. uncorrelated random effects > 0).
- * Starting equation number of uncorrelated random effects, uncorrelated random group number, and column positions from original data of each uncorrelated random effect by trait (if no. uncorrelated random effects > 0).

* Maximum number of repeated records by trait

MTDF51 - recoded data for MTDFRUN in binary format

MTDF52 - summary for each animal by record in binary format

MTDF66 - program log that includes summary statistics and order and format of MME

Prior to running MTDFPREP, make sure that any previous output files to be saved from MTDFPREP are renamed or copied elsewhere, e.g., MTDF66. MTDFPREP will delete or overwrite output files written in earlier runs of MTDFPREP.

MTDFRUN: Estimating Variance Components and Solving MME

MTDFRUN has several options based on the MME and produces corresponding output -- (co)variance components, solutions for covariates, fixed, and random effects, sampling variances and standard errors of solutions and contrasts, and expectations of solutions.

The first two interactive questions in MTDFRUN to be answered are :

TYPE OF ANALYSIS :

Is this a continuation of previous run?

For variance components chose (continue, yes = 1) to read in previous and existing simplex and continue iteration to improve local convergence or chose (n = 0, for start or restart) to construct new simplex and look for global maximum

For solutions to MME and standard errors of solutions chose (continue, yes = 1) to use previous final estimate from variance component estimation to build MME or chose (no = 1, for start or restart) to enter variance components for building MME.

and

OPTION FOR THIS RUN :

1 . . . iterate for variance components

2 . . . solutions for MME only

3 . . . solutions for sampling variances only

4 ... solutions for MME then sampling variances

For expectations of solutions use option 3 or 4;

After contrasts are completed, you will be

asked if expectations are wanted.

The rest of this section is divided into two parts. The first part is devoted to beginning an analysis. The second part is devoted to continuation of an analysis.

Initial run or cold start of MTDFRUN

This option should be used when a new analysis is being started for the first time or as a fresh restart to check if the log likelihood is the global maximum and not a local maximum for variance component estimation. The MTDF58 file, which contains the SPARSPAK reordering for a particular set of MME may or may not be present.

One question is :

No. of constraints for this analysis?

SPARSPAK requires that the MME be full rank. If equations corresponding to fixed effects need to be constrained to make the MME full rank, two answers are allowed. The actual number of constraints can be put in and then a prompt for each equation number to constrain will appear. However, too many constraints must not be imposed because this may lead to incorrect solutions. Alternatively, a 0 can be entered. If a 0 is entered and model dependencies are found, modifications in the SPARSNG subroutine (S. D. Kachman, Chapter 6) find MME dependencies and constrain those solutions to 0. After convergence has been declared, zeroes in the solution vector identify which equations were set to 0 and thus, the model dependencies.

Two additional interactive questions are asked that are apply only to OPTION 1 :

The number of parameters to hold constant?

This question is asked immediately after entering and verifying starting values (priors) for (co)variance component. This option is used to hold particular (co)variance components that are not 0, constant during the analysis. If a (co)variance is put in as 0, it will not move from 0 and is automatically held constant. This option may be useful if priors are not known for multiple trait

covariance(s) and good priors are desired for variances (see Chapter 3).

The other interactive question that applies to OPTION 1 is :

convergence criterion: minimum $V(-2\log L)$.

A convergence criterion of 1.e-6 to 1.e-8 is usually used.

OPTIONS 3 and 4 allow calculation of diagonal blocks of the inverse of the coefficient matrix of order up to 40 by 40 and of sampling variances and standard errors of individual solutions or contrasts. Inverse blocks, sampling variances, contrasts and standard errors of contrasts are written to MTDF67. Expectations of fixed effects can then be calculated and are also written to MTDF67. With OPTION 3, standard errors of contrasts can be obtained from the coefficient matrix without right hand sides but contrasts cannot be calculated because MME solutions are not obtained. Estimates of contrasts and their standard errors can be obtained with OPTION 4.

Running MTDFRUN

1. Compile MTDFRUN and MTDFLIK. PARAM.DAT needs to be available and the SPARSPAK or FSPAK files and MTDFSUB need to be linked.
2. Run MTDFRUN. The program reads MTDF44, MTDF50, MTDF51 and MTDF52. The program needs MTDF21 and/or MTDF22 if merging of solutions and labels are requested. Answers to the following questions are needed for the respective options:

OPTION 1	questions a through u
OPTION 2	questions a through s
OPTION 3	questions a through s and v through kk
OPTION 4	questions a through s and v through kk

- a) Description of analysis (up to 6 lines, terminated with a * in column 1 after last comment line).
- b) Is this a continuation or restart from a previous analysis : 0, no; 1, yes.
- c) Option for this run: 1, variance components; 2, solutions for MME; 3, sampling and contrast variances; 4, solutions for MME, contrasts, sampling and contrast variances and expectations of solutions.
- d) Number of constraints: can be 0 if Kachman modification to SPARSPAK or FSPAK is used. During Choleski factorization dependencies are found and solutions for dependent equations are set to 0. Alternatively, the program asks for equations to constrain. If the number of constraints is > 0 then equation numbers corresponding to fixed effects are requested (one equation number per line).
- e) Have the MME been reordered (MTDF58 must exist) : 0 no; 1 yes

- f) Input of priors for additive and second animal (i.e., the G_0 matrix) (co)variances. A display comes up according to the model described in MTDFPREP. With additive and second animal effects, additive effects for all traits followed by second animal effects will be displayed. The numbers following the effects, represent the trait. For example:

	a1	a2	m1
a1	1		
a2	2	4	
m1	3	5	6

Enter the position number and then the prior (co)variance corresponding to the position.

For example, for σ_{a1}^2 , an entry is :

1 100.d0 <return> [1 100 <return> or 1 100. <return> also work].

For σ_{A1A2} ;

2 -25.d0 <return>.

The matrix is initialized to zero. A prior is needed for each component to be estimated (0 is a valid estimate for covariances. If a covariance is put in as 0, or if nothing is entered in that position, it will remain 0 during the entire analysis.). If an error has been made, type -1 0.d0 <return> to show the position numbers again. Once all priors are entered, end the input by typing 0 0.d0 <return> [0 0 <return> also works].

- g) The priors will redisplay and verification is requested : 0, no; 1, yes.
- h) The number of parameters to hold constant. Only applicable to OPTION 1. Specific genetic (co)variances can be held constant while estimating other (co)variance components. If the answer is > 0, the prompt requests the position number(s) of the priors to be held constant.
- i) Input of priors for uncorrelated random effects, if applicable. The format is the same as for question f). If the same uncorrelated random effect is used across traits, a prior for the covariance across traits can be entered. However, with more than one uncorrelated trait present for a trait, there is no covariance between the uncorrelated effects within trait (enter as 0 or make no entry for that position). The program will not allow non-zero covariances for effects not coded in the same column in the data set. Some effects coded in the same column may still need to be zero. For example, if data include traits for scrotal circumference measured on bull calves and pelvic width measured on female calves, there would be no information to estimate a maternal permanent environmental covariance for these two traits, and the covariance should be restricted to zero. End input with 0 0.d0 <return>.

The headings for the covariances correspond to the trait (T) and the column (C) of

data which codes for the uncorrelated random effect. For example, if there were two traits and both traits had two uncorrelated random effects, with the effects coded for the first trait in columns 3 and 7, and the uncorrelated random effects for the second trait coded in columns 3 and 4, the display would appear as follows:

	T1 C3	T1 C7	T2 C3	T2 C4
T1 C3	1			
T1 C7	2	5		
T2 C3	3	6	8	
T2 C4	4	7	9	10

Non-zero covariances cannot be entered for elements: 2, 4, 6, 7, 9

- j) The priors for uncorrelated random effects will redisplay and verification is requested: 0, no; 1, yes.
- k) Number of parameters to be held constant for uncorrelated random effects. See h).
- l) Input of priors for residual (co)variances. See question f). End input with 0 0.d0 <return>
- m) The priors for residual variances will redisplay and verification is requested : 0, no; 1, yes.
- n) Number of parameters to be held constant for residual error effects. See h).
- o) Write covariate and fixed effect solutions to file MTDF77: 1, yes; 2, no.
If question o) is answered no, go to q)
- p) Merge labels with solutions for covariates and fixed effect to file MTDF77: 1, yes; 2, no. Note that the option for writing labels must have been requested in MTDFPREP to use this option.
- q) Write animal and second animal solutions to file MTDF78: 1, yes; 2, no.
- r) Write uncorrelated random effect solutions to file MTDF79: 1, yes; 2, no.
If question r) is answered no, go to t)
- s) Merge labels with solutions for uncorrelated random effect to file MTDF79: 1, yes; 2, no. Note that the option for writing labels must have been requested in MTDFPREP to use this option. If option 4, will solve and resume questions at v)
If option 1, will ask t) and u)
- t) Convergence criterion value. A suggestion is 1.e-6 to 1.e-8.
- u) Number of simplex rounds. Depending on model complexity and the data structure, it may take 1000 simplex rounds or more to achieve convergence. To obtain timings, enter 1 to obtain the initial SIMPLEX. The simplex will contain one more than the number of parameters to be estimated,

**** MME will now reorder and solve ****

**** and iterate until t) or u) is satisfied and option 1 will stop ****

with option 4, questions resume

- v) Calculate block of inverse; 0, no; 1, yes

- w) Enter the first and last rows for equations in block of inverse. The number of rows must be ≤ 40 . The prompts ask for the first equation number <return> and the last equation number <return>.
- x) After the block of the inverse has been calculated, do you want another block of the inverse to be calculated? 0, no; 1, yes. Question w) is repeated if answer is yes.
 *** Questions w) and x) will be repeated until no more block inverses are requested.
- y) Calculate the variance of a contrast? 0, no; 1, yes.
 *** if question y) is answered no, will go to question cc)
- z) Enter the number of elements in contrast vector, e.g., 2.
- aa) For the number of elements in the contrast vector, enter the row and contrast coefficient, one element and one coefficient per line <return>; (e.g., 4 1. <return>, 5 -1. <return>).
- bb) Calculate another contrast? 0, no; 1, yes.
 *** if question bb) is yes, repeat questions z - bb
- cc) Calculate PEV and $r_{t,i}$ for consecutive animals and sequential traits? 0, no; 1, yes. Note that $r_{t,i}$ will not be correct for group solutions or for PBV unless group solutions are estimated perfectly.
 *** if question cc) is no, will go to hh)
- dd) Enter equation for first trait on first animal to be considered.
- ee) Enter equation for **first** trait on **last** animal to be considered.
- ff) Enter the first trait wanted for each animal.
- gg) Enter the last trait wanted for each animal. Note for ff) and gg) that second animal effects are considered traits and can be included in list of traits - recall that the animal effects are sequential for all traits followed by second animal effects.
- hh) Calculate expectations for fixed effects; 0, no; 1, yes
 *** if question hh) is no, finished
- ii) Enter the first and last equation numbers for fixed effect parameters in expectations. Difference should not be greater than the total number of fixed effect levels. This answer will determine the number of rows in the matrix that will be printed out. The prompt asks for the equation number of the first parameter and the equation number for the last parameter.
- jj) Enter the first and last equation numbers representing solutions in block for expectations. The difference in these blocks cannot be larger than 40. A prompt asks for equation number for the first solution and the equation number for the last solution. [Expectations are relatively slow].
- kk) Do you want to calculate another set of fixed effect expectations? 0, no; 1, yes.
 *** if answer to kk) is yes, questions ii) and jj) will be repeated.

End of questions.

Results and Output Files from MTDFRUN by OPTION

Results/ File	OPTION 1	OPTION 2	OPTION 3	OPTION 4
Variance components	✓			
MME solutions	✓	✓		✓
Sampling and contrast variances and std errors			✓	✓
Contrasts				✓
Expectations of solutions for fixed effects			✓	✓
MTDF4 - answers to interactive questions that can be copied to TDFR5.DAT to use for a cold restart	✓	✓	✓	✓
MTDF54 - final simplex information	✓	(empty)	(empty)	(empty)
MTDF58 - reordered MME	✓	✓	✓	✓
MTDF59 - constraint information (if > 0)	✓	✓	✓	✓
MTDF67 - contrasts, sampling variances and expectations	(empty)	(empty)	✓	✓
MTDF68 - simplex history by round	✓	(empty)	(empty)	(empty)
MTDF72 - predicts BV and SEP	(empty)	✓	(empty)	✓
MTDF76 - program log	✓	✓	✓	✓
MTDF77 - MME solutions for covariates and fixed effects	✓	✓	(empty)	✓
MTDF78 - MME solutions for animal and 2nd animal effects	✓	✓	(empty)	✓
MTDF79 - MME solutions for uncorrelated random effects	✓	✓	(empty)	✓

Continuation of previous run

The continuation option can be used if the convergence criterion for (co)variance estimation has not been met, or if solutions, contrasts and standard errors are wanted after convergence is achieved. This option allows use of previous (co)variance components and constraint information.

For continuation with all options, MTDF54, which contains SIMPLEX information when the program finished previously, is needed to restart the program. The previous log file MTDF76 will be overwritten, so it should be renamed or copied elsewhere if it is to be saved. The user would usually print MTDF76 on completion of MTDFRUN. Files MTDF44, MTDF50, MTDF51, MTDF52, MTDF54, MTDF58, MTDF59 and MTDF68 must be available to the MTDFRUN program. MTDF21 and MTDF22 must be available if solutions are to be matched with labels from MTDFPREP.

Continuation with MTDFRUN

Run MTDFRUN. The program reads in MTDF44, MTDF50, MTDF51, MTDF52, MTDF54, MTDF58, MTDF59 and MTDF68. Answers to the following questions are needed for the respective options:

- a) Description of analysis (up to 6 lines, terminated with a *).
- b) Is this a continuation or restart from a previous analysis: answer 1, yes
- c) Option for this run: 1, variance components; 2, solutions for MME; 3, sampling and contrast variances; 4, solutions for MME, contrasts, and sampling and contrast variances, and expectations of solutions.
- d) Write covariate and fixed effect solutions to file MTDF77: 1, yes; 2, no.
If question d) is answered no, go to f)
- e) Merge labels with solutions for covariates and fixed effects: 1, yes; 2, no. Note that the option for writing labels must have been requested in MTDFPREP to use this option.
- f) Write animal and second animal solutions to file MTDF78: 1, yes; 2, no.
- g) Write uncorrelated random effect solutions to file MTDF79: 1, yes; 2, no.
If question g) is answered no, go to i)
- h) Merge labels with solutions for uncorrelated random effects: 1, yes; 2, no. Note that the option for writing labels must have been requested in MTDFPREP to use this option.
For option 4, skip to v) on page 19
For option 1, resume with i) and j).
- i) Convergence criterion value.
- j) Number of simplex rounds.

Convergence???

Even though an analysis converges to the desired convergence criterion, it is essential that the converged values be used as priors and a cold restart of MTDFRUN performed because the MTDFREML algorithm sometimes will converge to a local, rather than a global maximum. If MTDFRUN is restarted and the FVALUE becomes *smaller*, the analysis has moved to a better optimum. If two or three cold restarts converge to the same FVALUE, the global maximum has probably been found. To perform a cold restart, answer interactive questions as for an Initial Start or copy MTDF4 to MTDFR5.DAT and use for restart (see below). A cold restart is not a continuation of a previous analysis. Decrease magnitude of those covariances that yield correlations close to 1 or -1. Otherwise many non-permissible likelihoods may result.

Starting or Restarting MTDFRUN

On DOS or UNIX based systems, after gaining familiarity with the program, users may want to put the analysis information in a file for the program to read, which is easier than entering the data interactively. For restarts, users can copy MTDF4 to a restart input file. MTDF4 contains the original answers to the interactive questions except for the (co)variances, which are those obtained at the end of the previous run. However, please enter the data interactively to become familiar with the questions the first few times. If a mistake is made answering questions interactively, the program must be started from the beginning. To run the program using such an input file execute the program using the form:

```
mtdfrun.exe < input.fil
```

where mtdfrun.exe is the executable form of the MTDFRUN Fortran file and input.fil contains the same entries that would be entered interactively. If running the programs from a batch or script file it may be useful to use the command:

```
mtgsprep.exe < input.fil > output.fil
```

where output.fil is a file containing the prompts usually written to the terminal. An alternative approach is to change the file definition section for unit 5 in MTDFRUN to a physical file rather than keyboard input (i.e., change the value for IUN5 and add an open statement for that file).

CHAPTER TWO: Illustrations for MTDFREML

This chapter demonstrates models and analyses that can be run using MTDFREML. The examples presented are based on the mouse data distributed with DFREML (Meyer, 1991). Data are available on diskette or by anonymous FTP. The data files are distributed with example single and multiple trait MTDFREML analyses. The format of the pedigree file, MOUSE.PED, includes three fields: animal, sire, and dam. The data file, MOUSE.DAT, contains ten fields of data - seven integer and three real. The integers correspond to: animal, sire, dam, generation, sex, litter size (to use as a class factor), and litter number. The three real fields represent: litter size (for use as a covariate), body weight, and feed intake.

The purpose of this section is to illustrate interactive sessions with the programs and the types of output generated as well as what to examine and expect from the output. All analyses demonstrated here were run on 486/33 or Pentium class microcomputers with 32 or 64 MB of memory.

MTDFNRM

MOUSE.PED was the file used by MTDFNRM to produce the non-zero A^{-1} elements used in MTDFRUN. Two of the most important lines to note in the output file, MTDF56, are the number of pedigree lines read and the total number of different animals which is needed in MTDFPREP.

Results in MTDF56:

Started 09:38:18.52 on 03/01/1995

```

+++++
PROGRAM "MTDFNRM" - Calculate A-1 for "MTFRUN" and recode IDs for "MTDFPREP"
                   Version to use Westell grouping strategy
+++++

```

```

OPTION FOR CALCULATION OF A-1
  FOR ANIMAL   SIRE   DAM   TYPE ....  0
  FOR ANIMAL   SIRE   MGS   TYPE ...   1
OPTION CHOSEN FOR THIS ANALYSIS      =                0 see note 1
MAXIMUM ID                            =             41615
MINIMUM ID                             =                1
PEDIGREE FILE OPENED, IUN33           = mouse.ped
REORDERED ANIMAL FILE OPENED, IUN11   = MTDF11
FILE FOR A-1 ELEMENTS OPENED, IUN44   = MTDF44

```

```

FILE FOR IDS AND INBREEDING COEFFICIENTS OPENED
  THIS FILE WILL CONTAIN ANIMAL, SIRE, AND DAM
  RECODED AND ORIGINAL IDS FOLLOWED BY THE
  INBREEDING COEFFICIENT FOR EACH

```

```

NO. INTEGER FIELDS PER RECORD IN IUN33      =                4
ANIMAL ID IN POSITION .....                  1
SIRE ID IN POSITION .....                    2
DAM (MGS) ID IN POSITION ...                  3
NO. OF GENETIC GROUPS FOR CALCULATION OF W  =                0
The current time is: 09:38:18.74

```

```

NO. OF PEDIGREES READ                      =                309 see note 2
NO. OF DIFFERENT ANIMALS                   =                329 see note 3
INCLUDES NO. OF GENETIC GROUPS             =                0

```

```

END OF FIRST PASS
The current time is: 09:38:18.85

```

```

END OF SORT
The current time is: 09:38:18.85

```

```

FIRST 10 REORDERED IDs      1      215      see note 4
FIRST 10 REORDERED IDs      2      403
FIRST 10 REORDERED IDs      3      615
FIRST 10 REORDERED IDs      4      701

```

Note 1: The answers highlighted in gray were answers to the interactive question asked by MTDFREML. Check to make sure that they are correct

Note 2: Does this agree with your data? This number should equal number of data lines in pedigree file. Animals can be repeated in data file.

Note 3: This is the number of animals plus the number of base animals. Make sure that the number of base animals is at least 0. The number of base animals is the number of different animals minus the number of pedigrees read.

Note 4: Reordered animal identification numbers with original animal identification. These animal IDs should be reasonable given the data set.

FIRST 10 REORDERED IDs	5	814
FIRST 10 REORDERED IDs	6	904
FIRST 10 REORDERED IDs	7	1314
FIRST 10 REORDERED IDs	8	1602
FIRST 10 REORDERED IDs	9	1701
FIRST 10 REORDERED IDs	10	1813

ID VECTOR WRITTEN IN ORDER TO IUN11

The current time is: 09:38:18.90

SIRE AND DAM IN PEDIGREE REORDERED IN IVECS AND IVECD

The current time is: 09:38:18.96

CALCULATION OF A-1 FROM ANIMAL SIRE DAM (IOPT = 0)

NON-ZERO HS ELEMENTS FOR NRM INVERSE	=	1241	
LOG DETERMINANT OF NRM	=	-210.71674289	
NUMBER OF INBRED ANIMALS	=	0	
... WITH AVERAGE INBREEDING COEF	=	.00000000	
TOTAL NO. OF ANIMALS INCLUDING BASE			see note 3
AND GENETIC GROUPS	=	329	

Variance Component Estimation

Single Trait Model

The data for mouse body weight were analyzed with a model including additive (direct) genetic effect, correlated second animal genetic effect and one uncorrelated random effect. The data include 284 observations for body weight in mice. Additive direct genetic effect of animal, maternal genetic effect of second animal (the dam) and a maternal permanent environmental effect are in the model. Three fixed effects were: generation, sex and litter size.

MTDFPREP

For this example, the option to write levels of information to MTDF66 for all fixed effects was enabled and for the uncorrelated random effects was disabled. The complete list of answers to the interactive questions follow.

mouse.dat	name of data file
Mouse data from Karin Meyer	
Single trait analysis of body weight	
*	end of comments
7	number of integers on each line of data file
3	number of reals on each line of data file
1	number of traits in analysis
body weight	name of trait 1

2	position of trait in vector of real values
0.0	value of missing observation for trait 1
0	number of covariates
3	number of fixed factors
generation	name for fixed factor 1
4	position of fixed factor 1 in vector of reals
1	write summary of fixed factor 1 levels to log file (MTDF66)
sex	name for fixed factor 2
5	position of fixed factor 2 in vector of reals
1	write summary of fixed factor 2 levels to log file (MTDF66)
litter size	name for fixed factor 3
6	position of fixed factor 3 in vector of reals
1	write summary of fixed factor 3 levels to log file (MTDF66)
1	position of animal effect in vector of integers
329	num. of animals in relationship matrix (from MTDFNRM)
1	include second animal effect
maternal genetic	name of second animal effect
3	position of second animal effect in vector of integers
1	number of uncorrelated random factors
maternal perman env	name of uncorrelated random factor
3	position of uncorrelated random factor in vector of integers
0	do not write summary of uncorrelated random factor to log
1	write labels for covariates and fixed factors to MTDF21
1	write labels for uncorrelated random factors to MTDF22

Results in MTDF66:

Started 16:03:57.20 on 03/01/1995

```

+++++
PROGRAM "MTDFPREP" - Setup W=X:Z matrix for MT-IAM
Last revised 07-29-94
+++++

```

Data set description :
 Mouse data from Karin Meyer
 Single trait analysis of body weight

No. of data lines in Unit 33	=	284	see note 5
No. of integer variables per record	=	7	
No. of real variables per record	=	3	
No. of traits	=	1	
No. of valid records	=	284	
No. of RECORDS in Unit 33	=	284	
No. of animals with valid records	=	284	
No. of animals in A-1	=	329	
Order of MME (before constraints)	=	712	

Note 5: Does this correspond to your data?

Results for trait 1 - body weight (position 2) **see note 6**
 No. of records = 284 (missing value: .0000 No. missing = 0)
 Trait Mean SD CV Min Max Std Min Std Max
 1 24.0687 3.30236 13.72 14.600 34.500 -2.87 3.16

No. of covariates = 0

No. of fixed effects = 3
 1: 3 levels for generation (MME rows: 1 - 3)
 Level Value No. % Mean
 1 1 93 32.75 23.724 **see note 6**
 2 2 84 29.58 23.063
 3 3 107 37.68 25.158

2: 2 levels for sex (MME rows: 4 - 5)
 Level Value No. % Mean
 1 1 150 52.82 22.656 **see note 6**
 2 2 134 47.18 25.650

3: 7 levels for litter size (MME rows: 6 - 12)
 Level Value No. % Mean
 1 1 11 3.87 26.609 **see note 6**
 2 2 41 14.44 23.722
 3 3 25 8.80 24.864
 4 4 36 12.68 24.028
 5 5 96 33.80 24.265
 6 6 45 15.85 24.333
 7 7 30 10.56 21.973

No. of animals in A-1 = 329 (MME rows: 13 - 341)

No. of 2nd animal effects = 1 **see note 7**
 1: 329 levels for maternal genetic (MME rows: 342 - 670)

No. of uncorrelated random effects = 1 **see note 8**
 1: 42 levels for maternal perman env (MME rows: 671 - 712)

 Order of MME (before constraints) = 712 **see note 9**

Fixed effects = 3						
Trait	No.	Name	Position	Levels	Rows	
1	1	generation	4	3	1 -	3
1	2	sex	5	2	4 -	5
1	3	litter size	6	7	6 -	12

Animal effects = 1						
Trait	No.	Name	Position	Levels	Rows	
1	1	Animal w/ full A-1	1	329	13 -	341

Note 6: Are these characteristics of your data reasonable?
 Note 7: An equation is created for the second animal effect for all animals
 Note 8: An equation is created for each level of an uncorrelated random effect
 Note 9: Check number of levels and positions of fields in integer vector for possible input errors and order of MME

2nd Animal effects =							1
Trait	No.	Name	Position	Levels	Rows		
1	1	maternal genetic	3	329	342 -	670	

Uncorrelated random effects =							1	see note 9
Trait	No.	Name	Position	Levels	Rows			
1	1	maternal perman env	3	42	671 -	712		

Files written:

MTDF21 (ascii): Labels for covariates and fixed effects
 MTDF21 (ascii): Labels for uncorrelated random effects
 MTDF50 (ascii): Model information
 MTDF51 (binary): Recoded W=X:Z elements
 MTDF52 (binary): W summary for each animal
 The elapsed time was: 00:00:00.33

MTDFRUN

Answers to the interactive questions asked by MTDFRUN:

Mouse data from Karin Meyer

Single trait analysis of Birth Weight

*

0 continuation: 0=no; 1=yes
 1 run option: 1) var comp; 2) MME sol's; 3) sampling var; 4) MME sol's, contrast, & samp var
 0 # constraints **see note 10**
 0 reordered: 0=no; 1=yes
 1 4. animal effect starting value (direct genetic variance)
 2 .5 animal effect starting value (genetic covariance of direct and maternal effects)
 3 1.5 animal effect starting value (maternal genetic variance)
 0 0 end of genetic (co)variance input
 1 values are correct: 0=no; 1=yes; 2=redisplay screen
 0 # parameters to hold constant
 1 .5 uncorrelated effect starting value (maternal permanent environmental variance)
 0 0 end of uncorrelated random (co)variance input
 1 values are correct: 0=no; 1=yes; 2=redisplay screen
 0 # parameters to hold constant
 1 2. residual effect starting value (residual variance)
 0 0 end of residual (co)variance input
 1 values are correct: 0=no; 1=yes; 2=redisplay screen
 0 # parameters to hold constant
 1 write fixed effect solutions: 0=no; 1=yes
 1 merge label information from MTDFPREP with solutions: 0=no; 1=yes
 1 write animal solutions: 0=no; 1=yes
 1 write independent random effect solutions: 0=no; 1=yes
 1 merge label information from MTDFPREP with solutions: 0=no; 1=yes
 1.D-6 convergence criterion
 300 # Simplex rounds

Note 10: Number of constraints was entered as 0 because the Kachman modifications of SPARSPAK will automatically find singularities.

Results in MTDF76:

Started 08:43:46.88 on 03/02/1995

```
+++++
PROGRAM "MTDFRUN" - Estimate Covariance Components for MT-IAM
                        Last revised      8/      5/      94
+++++
```

Mouse data from Karin Meyer
Single trait analysis of Birth Weight

Cold start, i.e., not a continuation of previous run
Run option 1: iterate for variance components

0 constraints imposed by user

(Co)variances in model:
No. in likelihood calculation = 5
No. to be held constant = 0
No. to be maximized = 5

Starting values for this run:

G matrix:
4.0 .5
.5 1.5

C matrix:
.5

R matrix:
2.0

```
-----
**          Using SPARSPAK-A          **
** Rel. 3 ANSI Double Precision ver. **
**          (C) Univ. of Waterloo 1/84 **
-----
```

** reordering called **
** reordering completed **
The elapsed time was: 00:00:00.11

** solve5 called **
** solve5 completed **
The elapsed time was: 00:00:00.33

SPARSPAK-A statistics..

see note 11

Time:
Ordering = .059 secs. (.001 mins.)
Total/Solution = .000 secs. (.000 mins.)
Allocation = .000 secs. (.000 mins.)
Factorization = .000 secs. (.000 mins.)
Solve = .000 secs. (.000 mins.)

Note 11: Information provided by commercial SPARSPAK version.

Storage:
 No. equations = 712
 Non-zero hs elements in MME = 9742
 Sparsity of MME = 3.564%
 Maximum storage required = 12304. (.094 MB)
 Size of storage array (MAXSA) = 250000 (4.922% used)

***** RESULTS FROM SIMPLEX *****

OPTIONS SET FOR THIS RUN:
 MAXIMUM NO. OF SIMPLEX ITERATES ALLOWED = 300
 MINIMUM VARIANCE OF FUNCTION VALUES IN SIMPLEX = .1000000000E-05

RESULTS FOR THIS RUN:
 NO. OF SIMPLEX ITERATIONS CARRIED OUT = 44
 NO. OF LIKELIHOODS EVALUATED = 79
 NO. OF NON-PERMISSABLE PARAMETER VECTORS = 0
 NO. OF FAILED CONTRACTIONS = 0
 VARIANCE OF SIMPLEX FUNCTION VALUES = .2190791759E-06

Convergence criterion attained **See note 12**

Final Simplex: (++ best L; *** parameter held constant) **See note 13**

1	753.7488165	3.6828	.5114	1.3544	.4240	2.2586
2	753.7498457	3.6449	.4895	1.4246	.3877	2.2988
3	753.7487708	3.7863	.5039	1.3423	.4449	2.2142
4	753.7490006	3.6725	.4887	1.3914	.4260	2.2912
++ 5	753.7486078	3.6407	.4745	1.3861	.4140	2.2980
6	753.7494189	3.7354	.5228	1.3221	.4393	2.2565
-2 log L =		753.7486077658	(5)	Var =	.0000002191	

Estimates: **see note 14**

GENETIC VARIANCES AND COVARIANCES :

	a1	m1
a1 :	3.64066	.47452
m1 :	.47452	1.38608

UNCORRELATED RANDOM VARIANCES AND COVARIANCES:

	T1 C3
T1 C3 :	.414033

ENVIRONMENTAL VARIANCES AND COVARIANCES :

	e1
e1 :	2.29804

PHENOTYPIC VARIANCES AND COVARIANCES :

	p1
p1 :	8.21333

Note 12: Was convergence criteria met?

Note 13: The table includes the final simplex. The number of lines in the simplex is one more than the number of parameters to be estimated. The values in the simplex are the FVALUE, which is $-2\text{LOG}(\text{Likelihood})$ followed by the variance components in order of entry (i.e., genetic 1, genetic 2, ..., genetic g, uncorrelated random 1, ..., uncorrelated random u, residual 1, ..., residual r. The smallest FVALUE, which corresponds to the largest likelihood value, is given as well as the variance of the FVALUES, which is the convergence measure.

Note 14: The following estimates are based on the set of values with the smallest FVALUE (largest likelihood).

HERITABILITIES AND GENETIC CORRELATIONS

 a1 m1
a1 : .44
m1 : .21 .17

UNCORRELATED RANDOM EFFECTS PROPORTION OF TOTAL VARIANCE ON DIAGONALS,
CORRELATIONS ON OFF DIAGONALS

 T1 C3
T1 C3 : .504098E-01

ENVIRONMENTAL PROPORTION OF TOTAL VARIANCE AND CORRELATIONS

 e1
e1 : .28

0 Constraints:

see note 15

Files written:

MTDF4 (ascii): Parameter file (IUN5) for "cold" restart
MTDF54 (ascii): Last simplex
MTDF58 (binary): SPARSPAK reordering
MTDF59 (ascii): Constraints imposed
MTDF68 (ascii): Likelihoods by rounds
MTDF67 (ascii): Sampling variances if requested
MTDF72 (ascii): Predicted BVs and PEVs if requested
MTDF76 (ascii): Program log file
MTDF77 (ascii): Solutions for covariates and fixed effects if requested
MTDF78 (ascii): Solutions for trait within animal if requested
MTDF79 (ascii): Solutions for independent random effects if requested

The current time is: 08:44:06.82

Total time of analysis

The elapsed time was: 00:00:19.45

Note 15: The number of constraints will be 0 unless constraints were imposed.
The program does not count the number of constraints imposed by the
Kachman modifications.

Multiple Trait Model

The data for mouse body weight and feed intake were analyzed with a multiple trait animal model. The model for body weight included additive (direct) genetic effect and one uncorrelated random effect. The data include 284 observations for body weight. Additive direct genetic effect of animal and a litter effect are in the model. One covariate and one fixed effect were fit. Litter size was included as a covariate and generation was considered a fixed effect. The model for feed intake also included direct genetic effect and one uncorrelated random effect. There were two fixed effects included: litter size and generation.

Note that litter size is included as a covariate for one trait and a fixed effect for the other. The use of factor as a covariate and fixed effect is possible because there are two fields set for the same effect - one in the vector of integers and one in the vector of real values. If the same effect is not to

be used for two traits in the same analysis the same fields analysis and a fixed effect in a second analysis by placing the fields analysis and as an integer in the second analysis.

MTDFPREP

For this example, the option to write summary information to MTDF66 for levels of fixed factors was enabled for all fixed effects. The option was enabled for the uncorrelated random effect for body weight and disabled for feed intake. The complete list of answers to the interactive questions follow.

mouse.dat	name of data file	
Mouse data from Karin Meyer		
Multiple trait analysis of		
Body Weight and Feed Intake		
*	end of comments	
7	number of integers on each line of the data file	
3	number of reals on each line of the data file	
2	number of traits in the analysis	
Body Weight	name of trait 1	Trait 1
2	position of trait 1 in vector of real values	
0.	value of missing observation for trait 1	
1	number of covariates for trait 1	
Litter Size	name of covariate 1	
1	position of covariate in vector of real values	
1	maximum power of covariate	
1	number of fixed effects for trait 1	
Generation	name of fixed effect 1	
4	position of fixed effect 1 in vector of integers	
1	write summary of fixed effect 1 levels to log file (MTDF76)	
1	position of animal effect in vector of integers	
329	number of animals in relationship matrix (from MTDFNRM)	
0	include second animal effect for trait 1: 1=yes; 0=no	
1	number of uncorrelated random effects for trait 1	
Litter	name of uncorrelated random effect	
7	position of uncorrelated random effect in vector of integers	
1	write summary of uncorrelated random effect to log file (MTDF76)	
Feed Intake	name of trait 2	Trait 2
3	position of trait 2 in vector of reals	
0.	value of missing observation for trait 2	
0	number of covariates for trait 2	
2	number of fixed effects for trait 2	
Litter Size	name of fixed effect 1	
6	position of fixed effect 1 in vector of integers	

```

1          write summary of fixed effect 1 levels to log file (MTDF76)
Sex        name of fixed effect 2
5          position of fixed effect 1 in vector of integers
1          write summary of fixed effect 2 levels to log file (MTDF76)
0          include second animal effect for trait 1: 1=yes; 0=no
1          number of uncorrelated random effects for trait 1
Litter     name of uncorrelated random effect
7          position of uncorrelated random effect in vector of integers
0          do not write sum. of uncorr. random effect to log file (MTDF76)
1          write labels for covariates and fixed effects to MTDF21
1          write labels for uncorrelated random effects to MTDF22

```

Results in MTDF66:

Started 09:19:20.24 on 03/06/1995

```

+++++
PROGRAM "MTDFPREP" - Setup W=X:Z matrix for MT-IAM
Last revised 07-29-94
+++++

```

Data set description :
 Mouse data from Karin Meyer
 Multiple trait analysis of
 Body Weight and Feed Intake

```

No. of data lines in Unit 33          =          284
No. of integer variables per record   =           7
No. of real variables per record      =           3
No. of traits                         =           2
No. of valid records                  =          568
No. of RECORDS in Unit 33             =          284
No. of animals with valid records     =          284
No. of animals in A-1                 =          329
Order of MME (before constraints)     =          755

```

```

-----
Results for trait 1 - Body Weight          (position 2 )
No. of records = 284 (missing value: .0000 No. missing = 0 )
Trait  Mean      SD      CV      Min      Max      Std Min  Std Max
  1    24.0687    3.30236  13.72  14.600   34.500   -2.87    3.16

```

```

No. of covariates = 1
1: 1 regression coefficients for Litter Size (MME rows: 1 - 1)

```

```

Statistics for covariates:
Cov.  Mean      SD      CV      Min      Max      Std Min  Std Max
  1    4.47887    1.64829  36.80  1.0000   7.0000   -2.11    1.53

```

```

No. of fixed effects = 1
1: 3 levels for Generation (MME rows: 2 - 4)

```

Level	Value	No.	%	Mean
1	1	93	32.75	23.724
2	2	84	29.58	23.063
3	3	107	37.68	25.158

No. of animals in A-1 = 329

(MME rows: 14 -342)

No. of 2nd animal effects = 0

No. of uncorrelated random effects = 1

1: 42 levels for Litter

(MME rows:672 -713)

Level	Value	No.	%	Mean
1	1	8	2.82	23.800
2	2	7	2.46	23.014
3	3	5	1.76	22.880
4	4	7	2.46	24.129
5	5	8	2.82	21.687
6	6	8	2.82	18.300
7	7	7	2.46	25.471
8	8	7	2.46	27.186
9	9	7	2.46	23.514
10	10	8	2.82	28.375
11	11	7	2.46	22.943
12	12	6	2.11	23.467
13	13	8	2.82	23.750
14	14	8	2.82	24.212
15	15	4	1.41	25.400
16	16	8	2.82	19.113
17	17	6	2.11	27.317
18	18	8	2.82	23.850
19	19	6	2.11	24.750
20	20	6	2.11	25.067
21	21	4	1.41	21.925
22	22	6	2.11	21.033
23	23	5	1.76	24.180
24	24	8	2.82	23.537
25	25	7	2.46	22.386
26	26	8	2.82	19.462
27	27	7	2.46	24.843
28	28	7	2.46	25.429
29	29	8	2.82	25.175
30	30	8	2.82	22.950
31	31	7	2.46	25.457
32	32	8	2.82	23.450
33	33	6	2.11	23.983
34	34	2	.70	32.650
35	35	7	2.46	26.757
36	36	5	1.76	25.160
37	37	6	2.11	25.167
38	38	7	2.46	26.414
39	39	7	2.46	24.443
40	40	6	2.11	24.317
41	41	8	2.82	26.688
42	42	8	2.82	25.063

Results for trait 2 - Feed Intake

(position 3)

No. of records = 284 (missing value: .0000 No. missing = 0)

Trait	Mean	SD	CV	Min	Max	Std Min	Std Max
2	64.2556	5.93258	9.23	46.900	82.100	-2.93	3.01

No. of covariates = 0

No. of fixed effects = 2

1: 7 levels for Litter Size

(MME rows: 5 - 11)

Level	Value	No.	%	Mean
1	1	11	3.87	58.355
2	2	41	14.44	61.578
3	3	25	8.80	65.020
4	4	36	12.68	61.531

5	5	96	33.80	65.286	
6	6	45	15.85	66.242	
7	7	30	10.56	66.433	
2:	2 levels for Sex				(MME rows: 12 - 13)
Level	Value	No.	%	Mean	
1	1	150	52.82	61.392	
2	2	134	47.18	67.461	

No. of animals in A-1 = 329 (MME rows:343 -671)

No. of 2nd animal effects = 0

No. of uncorrelated random effects = 1
 1: 42 levels for Litter (MME rows:714 -755)

Order of MME (before constraints) = 755

Covariates =		1				
Trait	No.	Name	Position	Coeff.	Rows	
1	1	Litter Size	1	1	1 -	1

Fixed effects =		3				
Trait	No.	Name	Position	Levels	Rows	
1	1	Generation	4	3	2 -	4
2	1	Litter Size	6	7	5 -	11
2	2	Sex	5	2	12 -	13

Animal effects =		2				
Trait	No.	Name	Position	Levels	Rows	
1	1	Animal w/ full A-1	1	329	14 -	342
2	1	Animal w/ full A-1	1	329	343 -	671

Uncorrelated random effects =		2				
Trait	No.	Name	Position	Levels	Rows	
1	1	Litter	7	42	672 -	713
2	1	Litter	7	42	714 -	755

Files written:

- MTDF21 (ascii): Labels for covariates and fixed effects
- MTDF21 (ascii): Labels for uncorrelated random effects
- MTDF50 (ascii): Model information
- MTDF51 (binary): Recoded W=X:Z elements
- MTDF52 (binary): W summary for each animal

The elapsed time was: 00:00:00.38

MTDFRUN

Answers to the interactive questions asked by MTDFRUN.

Mouse data from Karin Meyer
 Multiple trait analysis of
 Body Weight and Feed Intake

*

```

0          continuation: 0=no; 1=yes
1          run option: 1 var comp; 2 MME sol's; 3 PEV; 4 MME sol's & PEV
0          # constraints
0          reordered: 0=no; 1=yes
1    7.87  animal effect starting value ( $\sigma_{A1}^2$ )
2    2.50  animal effect starting value ( $\sigma_{A1,A2}$ )
3    9.99  animal effect starting value ( $\sigma_{A2}^2$ )
0  0.D0    end of genetic (co)variance input
1          values are correct: 0=no, 1=yes, 2=redisplay
0          # parameters to hold constant
1    1.38  uncorrelated effect starting value ( $\sigma_{C1}^2$ )
2   -1.58 uncorrelated effect starting value ( $\sigma_{C1,C2}$ )
3    2.98 uncorrelated effect starting value ( $\sigma_{C2}^2$ )
0  0.D0    end of uncorrelated random (co)variance input
1          values are correct
0          # parameters to hold constant
1    2.66  residual effect starting value ( $\sigma_{R1}^2$ )
2    2.61  residual effect starting value ( $\sigma_{R1,R2}$ )
3   11.13 residual effect starting value ( $\sigma_{R2}^2$ )
0  0.D0    end of residual (co)variance input
1          values are correct
0          # parameters to hold constant
1          write fixed effect solutions: 0=no; 1=yes
1          merge label info. from MTDFPREP with solutions: 0=no; 1=yes
1          write animal solutions: 0=no; 1=yes
1          write uncorrelated random solutions: 0=no; 1=yes
1          merge label info. from MTDFPREP with solutions: 0=no; 1=yes
1.D-007   convergence criterion
200       # Simplex rounds

```

Results in MTDF76:

Started 13:15:33.10 on 03/06/1995

```

+++++
PROGRAM "MTDFRUN" - Estimate Covariance Components for MT-IAM
Last revised      8/          5/          94
+++++

```

Mouse data from Karin Meyer
Multiple trait analysis of
Body Weight and Feed Intake

Cold start, i.e., not a continuation of previous run
Run option 1: iterate for variance components

0 constraints imposed by user

```

(Co)variances in model:
  No. in likelihood calculation      =      9
  No. to be held constant           =      0
  No. to be maximized               =      9

```

Starting values for this run:

```

G matrix:
  7.9   2.5
  2.5  10.0

```

```

C matrix:
  1.4  -1.6
 -1.6   3.0

```

```

R matrix:
  2.7   2.6
  2.6  11.1

```

```

-----
**          Using SPARSPAK-A          **
** Rel. 3 ANSI Double Precision ver. **
**      (C) Univ. of Waterloo 1/84    **
-----

```

```

** reordering called **
** reordering completed **
The elapsed time was: 00:00:00.11

```

```

** solve5 called **
** solve5 completed **
The elapsed time was: 00:00:00.44

```

SPARSPAK-A statistics..

```

Time:
  Ordering      =      .113 secs. (      .002 mins.)
  Total/Solution =      .000 secs. (      .000 mins.)
  Allocation    =      .000 secs. (      .000 mins.)
  Factorization =      .000 secs. (      .000 mins.)
  Solve        =      .000 secs. (      .000 mins.)
Storage:
  No. equations      =      755
  Non-zero hs elements in MME =      14521
  Sparsity of MME    =      4.831%
  Maximum storage required =      16413. (      .125 MB)
  Size of storage array (MAXSA) =      250000 (      6.565% used)

```

***** RESULTS FROM SIMPLEX *****

OPTIONS SET FOR THIS RUN:

```

MAXIMUM NO. OF SIMPLEX ITERATES ALLOWED      =      200
MINIMUM VARIANCE OF FUNCTION VALUES IN SIMPLEX =      .1000000000E-06

```

RESULTS FOR THIS RUN:

```

NO. OF SIMPLEX ITERATIONS CARRIED OUT      =      161
NO. OF LIKELIHOODS EVALUATED              =      250
NO. OF NON-PERMISSABLE PARAMETER VECTORS   =      0

```

No. of failed contractions = 0
 Variance of simplex function values = .8790870688E-07
 Convergence criterion attained

Final Simplex: (++ best L; *** parameter held constant)

1	1963.666166	7.8101	1.6200	9.4198	1.3629	-1.6779	2.8546	2.7235	3.2129	11.7849
2	1963.665667	7.7670	1.5531	9.3912	1.3762	-1.6752	2.8575	2.7504	3.2503	11.7920
3	1963.666198	7.7990	1.5676	9.3751	1.3742	-1.6856	2.8497	2.7314	3.2409	11.8130
4	1963.666144	7.7988	1.5783	9.4316	1.3706	-1.6863	2.8664	2.7346	3.2539	11.8074
5	1963.665959	7.8109	1.6270	9.4488	1.3665	-1.6844	2.8526	2.7321	3.2280	11.7932
6	1963.665527	7.7794	1.5527	9.4424	1.3637	-1.6794	2.8355	2.7434	3.2502	11.7845
7	1963.666056	7.8120	1.5862	9.3669	1.3632	-1.6831	2.8688	2.7315	3.2407	11.8161
8	1963.665689	7.8089	1.6014	9.4493	1.3696	-1.6890	2.8582	2.7323	3.2326	11.7706
9	1963.665825	7.7782	1.5636	9.4284	1.3798	-1.6919	2.8523	2.7391	3.2533	11.7830
++10	1963.665334	7.7892	1.5675	9.3645	1.3739	-1.7060	2.8809	2.7346	3.2467	11.8274

-2 log L = 1963.6653344548 (10) Var = .0000000879

Estimates:

GENETIC VARIANCES AND COVARIANCES :

	a1	a2
a1 :	7.78922	1.56748
a2 :	1.56748	9.36453

UNCORRELATED RANDOM VARIANCES AND COVARIANCES:

	T1 C7	T2 C7
T1 C7 :	1.37390	-1.70597
T2 C7 :	-1.70597	2.88087

ENVIRONMENTAL VARIANCES AND COVARIANCES :

	e1	e2
e1 :	2.73456	3.24674
e2 :	3.24674	11.82745

PHENOTYPIC VARIANCES AND COVARIANCES :

	p1	p2
p1 :	11.89769	3.10825
p2 :	3.10825	24.07285

HERITABILITIES AND GENETIC CORRELATIONS

	a1	a2
a1 :	.65	
a2 :	.18	.39

UNCORRELATED RANDOM EFFECTS

PROPORTION OF TOTAL VARIANCE ON DIAGONALS, CORRELATIONS ON OFF DIAGONALS

	T1 C7	T2 C7
T1 C7 :	.115476	
T2 C7 :	-.857493	.119673

ENVIRONMENTAL PROPORTION OF TOTAL VARIANCE AND CORRELATIONS

	e1	e2
e1 :	.23	
e2 :	.57	.49

0 Constraints:

Files written:

MTDF4 (ascii): Parameter file (IUN5) for "cold" restart


```

MTDF54 (ascii): Last simplex
MTDF58 (binary): SPARSPAK reordering
MTDF59 (ascii): Constraints imposed
MTDF68 (ascii): Likelihoods by rounds
MTDF67 (ascii): Sampling variances if requested
MTDF72 (ascii): Predicted BVs and PEVs if requested
MTDF76 (ascii): Program log file
MTDF77 (ascii): Solutions for covariates and fixed effects if requested
MTDF78 (ascii): Solutions for trait within animal if requested
MTDF79 (ascii): Solutions for independent random effects if requested

```

The current time is: 13:17:11.09

Total time of analysis
The elapsed time was: 00:01:37.49

Contrasts, Sampling Variances and Expectations of Solutions

The multiple trait example is used to demonstrate calculation of contrasts, sampling variances and expectations of solutions for all of the fixed effects for both traits.. Because continuation rather than start (or restart) is chosen before doing option 4, MTDFRUN reads in the final simplex of the restarted run to use as parameters for the mixed model equations. If start or restart is chosen, a prompt to enter (co)variance components is given. Note that the option breeding values, prediction error variances, and r_{TI} should be used with caution, because it can take a great deal of time if requested for a large number of animals.

MTDFRUN

Answers to the interactive questions asked by MTDFRUN:

Mouse data from Karin Meyer
Multiple trait analysis of
Body Weight and Feed Intake

```

*
1      continuation: 0=no; 1=yes
4      run option: 1 var comp; 2 MME sol; 3 samp var; 4 MME sol, contrast, & samp var
1      write fixed effect solutions: 0=no; 1=yes
1      merge label information from MTDFPREP with solutions: 0=no; 1=yes
1      write animal solutions: 0=no; 1=yes
1      write uncorrelated random solutions: 0=no; 1=yes
1      merge label information from MTDFPREP with solutions: 0=no; 1=yes

```

```

1      calculate block of inverse: 0=no; 1=yes
1      first equation for block of inverse
8      last equation for block of inverse
0      calculate another block of inverse: 0=no; 1=yes
1      calculate the variance of a contrast: 0=no; 1=yes
1      number of elements in the contrast
1 1.   equation number of solution for contrast and coefficient for element 1
1      calculate the variance of another contrast: 0=no; 1=yes
2      number of elements in the contrast
4 1.   equation number of solution for contrast and coefficient for element 1
2 -1.  equation number of solution for contrast and coefficient for element 2
1      calculate the variance of another contrast: 0=no; 1=yes
2      number of elements in the contrast
13 1.  equation number of solution for contrast and coefficient for that solution
12 -1. equation number of solution for contrast and coefficient for that solution
1      calculate the variance of another contrast: 0=no; 1=yes
1      number of elements in the contrast
14 1.  equation number of solution for contrast and coefficient for that solution
0      calculate the variance of another contrast: 0=no; 1=yes
1      calculate predicted breeding values, prediction error variances, and rTI: 0=no; 1=yes
14     first equation for first animal for PBV, PEV, and rTI
18     first equation for last animal for PBV, PEV, and rTI
1      first trait for each animal
2      last trait for each animal
1      calculate the expected value of solutions: 0=no; 1=yes
1      first parameter number for expectation of solution
13     last parameter number for expectation of solution
1      first solution to calculate expectations
15     last solution to calculate expectations
0      calculate another set of expected value of solutions: 0=no; 1=yes

```

Results in MTDF76

Started 16:26:56.43 on 03/06/1995

```

+++++
PROGRAM "MTDFRUN" - Estimate Covariance Components for MT-IAM
                        Last revised      8/      5/      94
+++++

```

Mouse data from Karin Meyer
Multiple trait analysis of
Body Weight and Feed Intake

Continuation of previous run
Run option 4: solutions for MME, then sampling variances

Solutions only for this run:

RHS are for MME: X'R-ly
Z'r-ly

Estimates:

GENETIC VARIANCES AND COVARIANCES :

	a1	a2
a1 :	7.78922	1.56748
a2 :	1.56748	9.36453

UNCORRELATED RANDOM VARIANCES AND COVARIANCES:

	T1 C7	T2 C7
T1 C7 :	1.37390	-1.70597
T2 C7 :	-1.70597	2.88087

ENVIRONMENTAL VARIANCES AND COVARIANCES :

	e1	e2
e1 :	2.73456	3.24674
e2 :	3.24674	11.82745

PHENOTYPIC VARIANCES AND COVARIANCES :

	p1	p2
p1 :	11.89769	3.10825
p2 :	3.10825	24.07285

HERITABILITIES AND GENETIC CORRELATIONS

	a1	a2
a1 :	.65	
a2 :	.18	.39

UNCORRELATED RANDOM EFFECTS

PROPORTION OF TOTAL VARIANCE ON DIAGONALS, CORRELATIONS ON OFF DIAGONALS

	T1 C7	T2 C7
T1 C7 :	.115476	
T2 C7 :	-.857493	.119673

ENVIRONMENTAL PROPORTION OF TOTAL VARIANCE AND CORRELATIONS

	e1	e2
e1 :	.23	
e2 :	.57	.49

** Using SPARSPAK-A **
** Rel. 3 ANSI Double Precision ver. **
** (C) Univ. of Waterloo 1/84 **

** Reordering read from IUN58 **

** solve5 called **
** solve5 completed **
The elapsed time was: 16:26:57.31

0 Constraints:

SPARSPAK-A statistics..

Time:
Ordering = .113 secs. (.002 mins.)
Total/Solution = .000 secs. (.000 mins.)

```

Allocation      =      .000 secs. (      .000 mins.)
Factorization  =      .000 secs. (      .000 mins.)
Solve          =      .000 secs. (      .000 mins.)
Storage:
  No. equations      =      755
  Non-zero hs elements in MME =      14521
  Sparsity of MME    =      4.831%
  Maximum storage required =      16413. (      .125 MB)
  Size of storage array (MAXSA) =      250000 (      6.565% used)

```

```

Inverse block(s) written to iun67
Contrast information written to iun67
PBV, SEP, rTI written to iun72
Expectation of solutions written to iun67

```

Files written:

```

MTDF4 (ascii): Parameter file (IUN5) for "cold" restart
MTDF54 (ascii): Last simplex
MTDF58 (binary): SPARSPAK reordering
MTDF59 (ascii): Constraints imposed
MTDF68 (ascii): Likelihoods by rounds
MTDF67 (ascii): Sampling variances if requested
MTDF72 (ascii): Predicted BVs and PEVs if requested
MTDF76 (ascii): Program log file
MTDF77 (ascii): Solutions for covariates and fixed effects if requested
MTDF78 (ascii): Solutions for trait within animal if requested
MTDF79 (ascii): Solutions for independent random effects if requested

```

The current time is: 16:26:59.18

Total time of analysis
The elapsed time was: 00:00:02.31

Results in MTDF67:

Block of inverse for equations				1 to		8		
1	.0257	-.0030	.0035	.0067	.0085	.0150	.0139	.0033
2	-.0030	.6089	.4804	.4760	.0942	.0463	.0936	-.0086
3	.0035	.4804	.6988	.5743	.1039	.0988	.0919	.1027
4	.0067	.4760	.5743	.7513	.1158	.1127	.0889	.0911
5	.0085	.0942	.1039	.1158	4.4029	.5758	.7904	.5171
6	.0150	.0463	.0988	.1127	.5758	1.8164	.6410	.6922
7	.0139	.0936	.0919	.0889	.7904	.6410	2.7326	.5160
8	.0033	-.0086	.1027	.0911	.5171	.6922	.5160	1.8910

```

Contrast          1  with no. elements          1
Rows and coefficients for contrast
Row              1  coefficient          1.0000000000000000

```

```

Contrast:          1 = -4.537962624977266E-001
with SE = 1.604584640535971E-001

```

```

Contrast          2  with no. elements          2
Rows and coefficients for contrast
Row              4  coefficient          1.0000000000000000

```

Row 2 coefficient -1.0000000000000000

Contrast: 2 = 1.448625711542314
with SE = 6.388771261621322E-001

Contrast 3 with no. elements 2
Rows and coefficients for contrast

Row 13 coefficient 1.0000000000000000
Row 12 coefficient -1.0000000000000000

Contrast: 3 = 4.328618734541770
with SE = 4.548875893361996E-001

Contrast 4 with no. elements 1
Rows and coefficients for contrast

Row 14 coefficient 1.0000000000000000
Contrast: 4 = 1.568272269873763
with SE = 2.605093641450432

Contributions of parameters 1 to 13
To expectations for solutions 1 to 15

Parameter: Coefficients for solutions 1 to 15

PARAMETER	SOLUTION														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
13	0	0	0	0	1	1	1	1	1	1	1	-1	0	0	0

Results in MTDf72:

If genetic groups are used in the model the SEP are okay but the rTI are incorrect unless group effects are estimated perfectly!!!

Genetic variances used in order
7.789224982322628 9.364531657354942

animal	F	PBV	SEP	Rti	PBV	SEP	Rti	PBV	SEP	Rti
		PBV	SEP	Rti	PBV	SEP	Rti	PBV	SEP	Rti
		PBV	SEP	Rti	PBV	SEP	Rti	PBV	SEP	Rti
		PBV	SEP	Rti	PBV	SEP	Rti	PBV	SEP	Rti
		PBV	SEP	Rti	PBV	SEP	Rti	PBV	SEP	Rti
215	.00	1.568	2.61	.36	1.277	2.91	.31			
403	.00	1.568	2.61	.36	1.277	2.91	.31			
615	.00	1.345	2.60	.36	1.324	2.89	.33			
701	.00	1.345	2.60	.36	1.324	2.89	.33			
814	.00	-.500	2.64	.33	1.000	2.92	.30			

There are several things that should be noticed in these results. The variance components in MTDF76 should match the converged values from the previous run if continuation was chosen (see the results from the previous multiple trait analysis). There is little other useful information found in MTDF76 when the program is used for options other than variance component estimation. The block of inverse of the coefficient matrix is given in MTDF67. There were 4 contrasts evaluated corresponding to:

- 1) covariate of litter size for body weight,
- 2) difference of generations 3 and 1 for body weight,
- 3) difference of sex effects for feed intake, and
- 4) animal effect for the first animal and first trait with standard error of prediction.

By looking at the expectations of the solutions, it is clear that the covariate and generation effects are estimable for body weight, because there is only one fixed effect. However, the expectations for fixed effects of the second trait (feed intake) show that the solution includes the underlying parameter but also includes a function of the last level of the fixed effect for sex, which was constrained to zero. Therefore, in order to evaluate sex effects for feed intake, the difference must be used (in this case by using a one element contrast for equation 12, because equation 13 is constrained to zero: a two solution contrast can also be used 12, 1. and 13 -1 with the same result). The Kachman modifications constrained the last level of the fixed effect for sex to be zero in this case, but it is important to note that this will **NOT** always be the case. Because the equations are reordered by SPARSPAK or FSPAK, the user should not assume that the last level of second and later fixed effects are constrained to zero. You must still use complete estimable functions or check the estimability for a particular solution by using the expected values of the solutions as was done here. Finally, the use of expectations and contrasts is not limited to fixed effects. The expectations of solutions included two animal effects to demonstrate that for this analysis the animal solutions did not include any of the fixed effects. In addition, the last contrast was used to calculate the standard error of prediction of the breeding value for the first animal for body weight. The same information for genetic effects can also be obtained in MTDF72 for predicted breeding values, standard errors of prediction, and accuracies (r_{T1}) for a selected group of animal effects. It should be noted that these two methods yield identical results for the prediction and standard error, i.e., equation 14 is the equation for the animal effect for the first animal (original ID 215).

CHAPTER THREE: Alternative Models and Shared Experiences with MTDFREML

Sex-Limited Traits and Genetic-Environmental Interactions

The programs can accommodate traits for which measurements are available for a trait on one sex and for another trait on the other sex, e.g., scrotal circumference for males and age at first estrus for females. Such pairs of traits can be analyzed together with traits measured on both sexes. For the sex-limited traits, the ties between the traits are entirely from the relationship matrix, e.g., sires or dams in common for male and female progeny. Experience has shown a tendency in small data sets for the across-trait genetic correlations to converge to 1 or -1.

The pedigree file for MTDFNRM will include all animals. Fields for both traits must be present for each animal in the data file for use in MTDFPREP. For example, males will have fields for both scrotal circumference and age at first estrus (always filled with the missing observation indicator) and females will also have the same fields for both traits but with the field for scrotal circumference filled with the missing observation indicator. When running MTDFRUN, the environmental covariance between scrotal circumference and age at first estrus must be zero. With MTDFREML combinations of sex limited and other traits can be considered. Do not enter values for environmental covariances that are not estimable, e.g., pairs of traits never measured on the same animal.

Another application of the sex-limited type of analysis is to consider the same name trait to be different traits in males and females. The data file must have different fields for the male and female traits with one field with the designated missing value. Again, the environmental covariance will be zero.

Yet another similar application is for some progeny of bulls to be exclusively in one environment and other progeny to be exclusively in another environment. Environments might be breed of dam, locations; even male measures from one breed of dam and female measures from a different breed of dam. The combinations are nearly endless. Unfortunately, problems with convergence for small data sets with poor relationship structures may also be nearly endless.

Nevertheless, REML provides a unified method to attempt such analyses.

Sire Models

Although this set of programs may not be the most efficient for sire models because of increased density of the coefficient matrix, such models can be run, e.g., the usual sire model, the sire and maternal grandsire model, and the sire and dam model.

Sire model

For MTDFNRM, the pedigree file needs to be made for sires' having progeny with records. The sire, sire of sire, maternal grandsire of sire option would be chosen. The identification vector would contain:

sire instead of animal

sire of sire instead of sire of animal

maternal grandsire of sire instead of dam of animal

If some sires are related through females, e.g., are full sibs, then those connecting dams can be included in the pedigree file in which case the animal, sire, dam option would be used with sires and the connecting dams in the pedigree file:

sire	and	dam
sire of sire		sire of dam
dam of sire		—

In MTDFPREP when the field for animal is requested, the field for sire would be given.

The MTDFRUN program works with whatever model is specified. The output, however, is formulated in terms of an animal model. Therefore, for a sire model, heritabilities would need to be recomputed, i.e., four times the heritability stated on the output. Estimates of heritability for the animal model are forced to be less than or equal to one. With the program run as a sire model, heritability from four times the sire variance may exceed one. Solutions for sires will be for transmitting ability or one-half of predicted breeding value. Standard errors for sire solutions will be for prediction error for transmitting ability.

Sire and maternal grandsire model

For data sets with too many animals for using an animal model, an option may be a model including sire of the animal effect (.5 direct genetic of the sire) and maternal grandsire of the animal

(.25 direct genetic of mgs and .50 maternal genetic of mgs).

For MTDFNRM, the alternatives are as described for the sire model.

For MTDFPREP, the field for sire will designate the primary animal effect and the field for maternal grandsire will designate the second animal effect.

With MTDFRUN, the parameter space for estimates is for an animal model. Thus, estimates for sire and mgs variances and sire-mgs covariance will be forced to have a sire-mgs correlation in the usual bounds of -1 to +1. For example, with one trait the heritabilities and correlations calculated from

$$\begin{aligned}\sigma_s^2 &= \sigma_g^2/4 \\ \sigma_{mgs}^2 &= \sigma_g^2/16 + \sigma_{gm}/4 + \sigma_m^2/4 \\ \sigma_{s,mgs} &= \sigma_g^2/8 + \sigma_{gm}/4\end{aligned}$$

may not be in the parameter space. To estimate the sire-mgs covariance as many sires as possible should also be maternal grandsires. Similar considerations hold for the covariances and variances for two or more traits.

Sire and dam model

A sire and dam model depending on relationships among sires and dams may allow estimation of genetic direct-maternal covariance but will not allow breaking out a dominance component from the maternal variance.

For MTDFNRM, animals in the pedigree file will be sires and dams of animals with records and their ancestors. Use the option for A⁻¹ corresponding to whether the pedigree file is for animal, sire, and dam or (sire, sire of sire, and maternal grandsire of sire) and equivalent for dam.

sire	dam
sire of sire	sire of dam
dam of sire or maternal grandsire of sire	dam of dam or (—)

For MTDFPREP, enter either sire or dam as primary animal and the remaining parent as the second animal. Unfortunately, the way the program is set up, equations for a first animal and a second animal effect will be set up for all sires and dams. That should not affect the answers for either sire and dam effects or for variance components but more equations and coefficients will be

involved. Some of the sires and dams must be related for estimating the sire by dam covariance. The parameter space considered for MTDFRUN will bound the sire-dam correlation but not heritabilities and direct-maternal genetic correlations. For example, with one trait the expectations of the variance components:

$$\begin{aligned}\sigma_s^2 &= \sigma_g^2/4 \\ \sigma_d^2 &= \sigma_g^2/4 + \sigma_m^2 + \sigma_{gm}/2 \\ \sigma_{sd} &= \sigma_g^2/4 + \sigma_{gm}/2\end{aligned}$$

can be used to estimate direct and maternal heritability and the direct-maternal covariance. If some dams have more than one progeny, then including an uncorrelated random permanent environmental maternal effect probably is necessary. The field specified for PE-maternal is the same as for the dam. Note that records of animals that are also sires and dams can be included but relationships will not be considered correctly.

Models with Many Levels of a Fixed Factor

With sire or sire-mgs models, many levels of a fixed factor such as herd-year-season or contemporary group effects would be likely. Such effects can be argued to be fixed or random. If no association exists, e.g., between herd effects and sires, then such effects might be modelled as an uncorrelated random factor. Modelling the effects as fixed, however, adjusts for the associations but with the cost of making all effective comparisons within levels of that factor. Two suggestions for using the MTDFREML program follow but have not been tested for the factor considered fixed. First, the PARAM.DAT file can be changed to increase the maximum number of levels for each fixed factor, MAXNFL. In that case, the maximum number of traits, MAXTRT, and maximum number of fixed effects for any trait, MAXFIX, should be reduced to a minimum. These multiply together in creating memory requirements. If memory requirements are still too great, then a possibility is to sacrifice the ability to obtain expected values by reducing the size of the commented matrix in MTDFLIK to store expected values to a minimum but be sure not to ask for expected values.

A second possibility is one that seems to work and may be a practical compromise to the question of whether contemporary group effects are random or fixed. The compromise is to enter the field as an uncorrelated random effect in MTDFPREP. Then in MTDFRUN assign a small variance

for the effect that is held constant (with small depending on the situation; e.g., as small as .0001% of the phenotypic variance for the trait). The very small constant variance makes the factor "almost fixed".

Standard Statistical Models Without Relationships

Most statistical models do not have random factors with correlated levels, such as are considered with the inverse of the relationship matrix. Such models can be used with MTDFREML. Several possibilities exist; some would require program changes. The following seems to work. Create a pedigree file with no sire or dam identification. The animal identification would be the uncoded levels of one of the random factors. A field with only zeroes could be specified for both sire and dam in MTDFNRM. The A^{-1} elements would then correspond to those for an identity matrix. A second animal effect should not be specified because as with the sire and dam model, both a second and a first animal effect will be included for each "animal" (i.e., levels in the first and second animal fields). Other uncorrelated random factors can also be specified. Correlations among the uncorrelated effects for a trait should have priors set to zero.

Derivative-free algorithms that take advantage of sparse matrices have greatly expanded the magnitude of analyses that can be used to obtain REML estimates. The algorithms, however, can be just as frustrating and as susceptible to misuse as any statistical packages.

Local vs Global Maximization

The simplex algorithm is not guaranteed to converge to a global maximum (see e.g., Chapter 7). The variance of the simplex (the convergence criterion) depends entirely on the current simplex and if convergence is to a local maximum, that variance may become very small. MTDFRUN should ALWAYS be restarted with the estimates at apparent convergence as initial values. For estimates near zero, a small restarting value such as .0010 might be appropriate but not .0000. In some cases, further fresh (i.e., cold) restarts should be made. A future option to be added to MTDFREML will be for a cold restart with each parameter estimate in turn to be increased by a factor of .0002 and then also to be decreased by the same factor (O'Neill, 1971). If any FVALUE (-2 log likelihood) is smaller than the FVALUE at apparent convergence, the simplex procedure should be continued from the restart until the variance of the simplex is less than $1.e-9$. Second and later restarts may be necessary.

Starting values that are much too large or starting covariances with the wrong signs seem to increase number of rounds needed for convergence. Crossing over from a positive to a negative covariance is particularly slow. Movement in the simplex is a small fraction of the previous estimate. Thus, small covariances do not move very fast. Choose the variance component priors carefully. Do not choose covariance priors that give correlations close to -1 or +1 which will put the simplex immediately on a boundary.

Standardization of Data

Some discussion has indicated that the DFREML approach may be susceptible to rounding errors. Rounding errors seem more serious for analyses for traits measured on greatly different scales, e.g., milk yield and pelvic height. The different scales also can create difficulty in reading the MTDF76 file output from MTDFRUN. That problem can be alleviated by assigning appropriate format statements to some of the WRITE (IUN66,_) statements. [The MTDF68 file, however, will contain the FVALUE and parameter estimates for each likelihood evaluation in free format.] A reasonable procedure may be to divide each trait by an approximate standard deviation. At global convergence, the (co)variance estimates can be rescaled to the original units, e.g., variance multiplied by the square of the approximate standard deviation.

In one analysis of binomial data the FVALUE would not converge and, in fact, alternated between two quite different values. The variance components were quite small. Multiplication of the binomial measures by 100 resulted in convergence. Except for rounding error, scaling should not have affected minimizing the FVALUE. Nevertheless, this experience suggests that successful use of MTDFREML is not entirely "science" but is partly an "art".

Sparseness

MTDFREML is computationally efficient when the coefficient matrix is sparse. The inverse of the relationship matrix (A^{-1}) is sparse but fixed effects can create enough density to slow the SPARSPAK subroutines. Covariates will slow SPARSPAK the most. The covariate by covariate block of the coefficient matrix is completely filled as will be much of the block for the covariate by fixed classification factors.

Sparsity can be enhanced by using subclass models for fixed classification factors. For

example, the subclass block for herd by year effects will be diagonal but separate herd, year, and herd-year effects will create more fill and more equations. Unless residual degrees of freedom are a problem or unless solutions for main effects are needed, subclass models will be more efficient computationally. The contrast option in MTDFRUN allows obtaining main effect estimates from the subclass solutions after convergence. For larger data sets, pre-adjustment of data for effects such as sex and age of dam will also keep the coefficient matrix more sparse.

Contrasts

Contrasts and standard errors for fixed effects can be obtained relatively easily. That option does not check for estimability. For example, suppose the contrast is for year 2 plus year 3 effects. The expected value for the year 2 solution might be year 2 minus year 1 effects and for the year 3 solution the expected value would be year 3 minus year 1 effects. The contrast (probably not a good name when misused) would be calculated and the standard error of year 2 plus year 3 solutions would be calculated but would have little meaning unless the expectations were known. Therefore, a recommended procedure is to use the expected value option for each solution used in a contrast.

Number of Traits

Running more than 3 or possibly 4 traits at a time is probably not a good idea, especially with complex models, e.g, several correlated random effects. The number of evaluations for convergence will be considerable due to the many parameters to estimate. Covariance components that go to correlation boundaries and variances that go to zero markedly increase the number of likelihoods to be calculated. With many traits and components some of these situations are likely. A useful strategy might be the following.

Strategy for Estimation of Covariances with Multiple Traits

The following strategy for complex multiple trait models may be useful.

- 1) Run each trait separately to convergence at moderate level of convergence, e.g., 1.e-6 by setting all covariances = 0.d0.
- 2) Cold start with variance estimates from single trait analyses held constant and use guessed covariances as starting values for multiple trait analyses.

- 3) Run multiple trait analysis to low level of convergence, e.g., 1.e-3 or 1.e-4.
- 4) Do not hold variances constant and cold restart from apparently converged estimates from 3). Run to low level of convergence, e.g. 1.e-3.
- 5) Repeat cold restarts until -2 log likelihood does not change in units (tens?) position at low level of convergence.
- 6) Then run to high level of convergence, e.g., 1.e-9.
- 7) To make sure of global convergence, cold restart from converged estimates. Run to high level of convergence. Repeat if a larger log likelihood (or small -2 log likelihood) is found. Changes in -2 log likelihood beyond the third decimal position are not usually important.

Insufficient Memory

An insufficient memory error may occur for a multiple trait model with numerous covariates and fixed effects with many levels. The matrix **EXPMAT** used for expectations of solutions would use a large amount of storage. Suppose an analysis has three traits, maximum number of covariates is three, maximum number of regression coefficients is two, maximum number of fixed effects is four with 2000 levels. Then the EXPMAT matrix requires 7506 MB of memory.

Two possible solutions require recompiling MTDFLIK and MTDFRUN:

- 1) Reduce maximums for number of traits, covariates, regression coefficients, fixed effects and levels in PARAM.DAT to the minimums needed.
- 2) Replace the EXPMAT matrix definition with EXPMAT(1,1). Then expectations of fixed effects and covariates can not be calculated.

CHAPTER FOUR: Theoretical Considerations for MTDFREML

Mixed Models

In matrix notation the general mixed model for an observation vector, \mathbf{y} , is:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \mathbf{e}, \text{ where}$$

$\boldsymbol{\beta}$ = vector of fixed effects associated with records in \mathbf{y} by \mathbf{X} , and

\mathbf{u} = vector of random effects associated with records in \mathbf{y} by \mathbf{Z} .

Now $E[\mathbf{y}] = \mathbf{X}\boldsymbol{\beta}$, but with $V(\mathbf{u}) = \mathbf{G}$, $V(\mathbf{e}) = \mathbf{R}$, and $E[\mathbf{ue}'] = \mathbf{0}$; $V(\mathbf{y}) = \mathbf{ZGZ}' + \mathbf{R}$, where some covariances among the y 's, \mathbf{ZGZ}' , are introduced by having random effects in common.

In a common animal breeding application for a single trait analysis, \mathbf{u} is a vector of breeding values with $V(\mathbf{u}) = \mathbf{G} = \mathbf{A}\sigma_g^2$, where \mathbf{A} is the numerator relationship matrix and σ_g^2 is the additive genetic variance (variance of breeding values) and $\mathbf{R} = \mathbf{I}\sigma_e^2$.

Henderson's Mixed Model Equations

Henderson's mixed model equations (e.g., 1950, 1963, 1975, 1984a) simplify for many situations the calculation of $\boldsymbol{\beta}$ and $\hat{\mathbf{u}}$. In general form the MME are:

$$\begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z} + \mathbf{G}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{y} \end{bmatrix}$$

Although \mathbf{R} is of order the number of records, \mathbf{R} is usually assumed to be diagonal for single trait analyses, often $\mathbf{I}\sigma_e^2$, and block diagonal (blocks of order of number of traits) for multiple trait analyses, so that calculations with \mathbf{R}^{-1} are easy. Henderson et al. (1959) proved the $\hat{\boldsymbol{\beta}}$ from his equations are BLUE as from generalized least-squares and Henderson (1963) proved the $\hat{\mathbf{u}}$ are BLUP.

MME for Variance Component Estimation

The use of MME for prediction of breeding values has become commonplace. The story of the MME for estimation of variance components may not be so well known. In a 1968 paper Cunningham and Henderson (1968) described an iterative procedure for estimating variances and fixed effects which may have been the forerunner of what is now known as restricted maximum

likelihood (REML), the current method of choice. In 1969, Thompson (1969) corrected the error in the denominator of the Cunningham and Henderson estimator of the residual variance (the number of random effects should not be subtracted in calculation of degrees of freedom). Two years later, Patterson and the same Thompson (1971) presented what is now known as REML. They maximized the part of the multivariate normal likelihood associated with the random effects after essentially adjusting for estimates of the fixed effects. Their derivation was for a function of \mathbf{y} , $\mathbf{K}'\mathbf{y}$, that has $E[\mathbf{K}'\mathbf{y}] = E[\mathbf{K}'\mathbf{X}\boldsymbol{\beta}]$ with the restriction, $\mathbf{K}'\mathbf{X} = \mathbf{0}$, so that the values of the variance components that maximize the likelihood are invariant to constraints to obtain $\hat{\boldsymbol{\beta}}$.

The logarithm of the restricted multivariate normal likelihood can be written as:

$$\Lambda = -.5[(n - p)\log(2\pi) + \log|\mathbf{K}'\mathbf{V}\mathbf{K}| + \mathbf{y}'\mathbf{K}(\mathbf{K}'\mathbf{V}\mathbf{K})^{-1}\mathbf{K}'\mathbf{y}] .$$

Although $\mathbf{K}'\mathbf{X} = \mathbf{0}$ guarantees invariance, the maximization of Λ does not require knowing \mathbf{K}' . A somewhat more familiar identity with constraints already imposed on \mathbf{X} is:

$$\Lambda = -.5[\text{constant} + \log|\mathbf{V}| + \log|\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}| + (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})] .$$

Note that $(n - p)\log(2\pi)$, the constant, is not affected by choice of \mathbf{V} to maximize Λ , where n is the number of records and p is the rank of the part of the coefficient matrix due to fixed effects. The familiar terms are:

- 1) $\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \mathbf{R}$, the variance of \mathbf{y} ,
- 2) $\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}$, the coefficient matrix for GLS estimation of $\hat{\boldsymbol{\beta}}$, and
- 3) the last term is the generalized residual sum of squares with residuals weighted by the inverse of \mathbf{V} . Harville (1977) and Searle (1979) developed an equivalent form of Λ that is important for derivative-free REML (DFREML):

$$\Lambda = -.5[\text{constant} + \log|\mathbf{R}| + \log|\mathbf{G}| + \log|\mathbf{C}| + \mathbf{y}'\mathbf{P}\mathbf{y}] \text{ where,}$$

\mathbf{C} is the full-rank coefficient matrix for the MME and

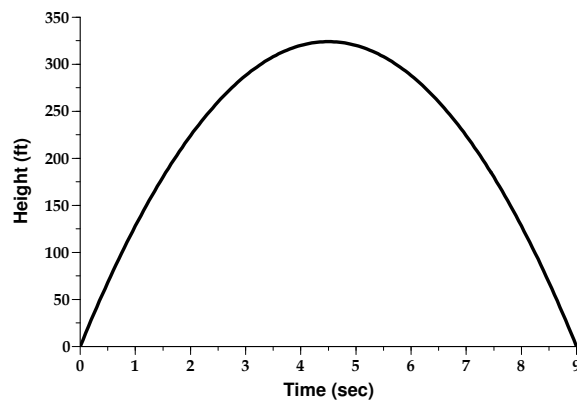
$\mathbf{y}'\mathbf{P}\mathbf{y}$ with $\mathbf{P} = \mathbf{V}^{-1} - \mathbf{V}^{-1}\mathbf{X}(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}$, although formidable in appearance, is the generalized residual sum of squares.

Obviously to work comfortably with Λ , a refresher course on determinants, particularly the logarithms of determinants is desirable. Although desirable, that can not be done here, but the patterns can be worked out easily with examples and Searle's (1982) matrix algebra book. Derivative methods such as EM-REML to obtain estimates of \mathbf{G} and \mathbf{R} involve derivatives of Λ with respect to unique variances and covariances in \mathbf{G} and \mathbf{R} which are non-linear in \mathbf{G} and \mathbf{R} and require non-linear

iteration.

Derivative-Free REML

Smith and Graser (1986) and Graser et al. (1987) broke with tradition and proposed not taking derivatives or expectations. The derivative-free method, very simply, is to try different \mathbf{R} and \mathbf{G} (e.g., σ_e^2 of $\mathbf{R} = \mathbf{I}\sigma_e^2$ and σ_g^2 of $\mathbf{A}\sigma_g^2$) until the combination that maximizes the log likelihood, Λ , is found for the data, \mathbf{y} . That simple approach is not quite as simple computationally as it first seems, but is perhaps simpler than approaches such as EM-REML involving derivatives, expectations, and inverses. A simple example will be used to demonstrate the difference between derivative and derivative-free approaches to maximizing a function. The example is not of maximizing a likelihood but describes finding the maximum of a function with a defined equation. The figure describes the problem. A baseball is thrown into the air at 144 ft/sec. Gravity pushes back. The equation for height above the ground is: $s = \text{function}(t = \text{time in sec}) = 144t - 16t^2$.



The figure is a plot of height and time. The plot itself actually shows the number of seconds when the maximum height is reached. Plotting all such points for complicated functions, however, would be inefficient. The derivative method of finding t to maximize s is to take the derivative of s with respect to t and equate to zero:

$$ds/dt = 144 - 32t = 0.$$

Thus $t = 144/32 = 4.5$ sec is obtained directly.

The derivative-free method is similar to making the plot. Values of t are put into the equation until the maximum is found. For a problem as simple as this one the logic for the search is fairly

straight forward: try a small value, a larger value, and then successively try to bracket the maximum until the maximum is found, as for example:

<u>step</u>	<u>t</u>	<u>s</u>
1	1.0	128.00
2	4.0	320.00
3	5.0	320.00
4	4.4	323.84
5	4.6	323.84
6	4.5	324.00

For estimating variance components the equation to be maximized based on the MME has already been described:

$$\Lambda = -.5[\text{constant} + \log|\mathbf{R}| + \log|\mathbf{G}| + \log|\mathbf{C}| + \mathbf{y}'\mathbf{P}\mathbf{y}] .$$

This form of the likelihood is completely general in \mathbf{R} and \mathbf{G} and the sample of records, \mathbf{y} . The important point is that \mathbf{R} and \mathbf{G} can be simple or complicated, for a single trait or multiple traits. As the structures of \mathbf{R} and \mathbf{G} become more complicated, the problem of searching efficiently for better sets of \mathbf{R} and \mathbf{G} becomes more and more difficult. In Λ the constant term can be ignored.

Often evaluating the log likelihood is less confusing if instead of maximizing Λ , -2Λ is minimized; i.e.,

$$-2\Lambda = \text{constant} + \log|\mathbf{R}| + \log|\mathbf{G}| + \log|\mathbf{C}| + \mathbf{y}'\mathbf{P}\mathbf{y} .$$

In the expression to be minimized, the constant is ignored. At each round the other four terms must be calculated. The easy terms are $\log|\mathbf{R}|$ and $\log|\mathbf{G}|$. For example, with $\mathbf{R} = \mathbf{I}_n\sigma_e^2$, $\log|\mathbf{R}| = n\log(\sigma_e^2)$ and with $\mathbf{G} = \mathbf{A}\sigma_g^2$, $\log|\mathbf{G}| = \log|\mathbf{A}| + q\log(\sigma_g^2)$ where q is the order of \mathbf{A} , the numerator relationship matrix. If, for a model with correlated direct and maternal genetic effects and permanent environmental effects with p levels and variance, σ_p^2 :

$$\mathbf{G} = \begin{bmatrix} \mathbf{A}\sigma_g^2 & \mathbf{A}\sigma_{gm} & \mathbf{0} \\ \mathbf{A}\sigma_{gm} & \mathbf{A}\sigma_m^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_p\sigma_p^2 \end{bmatrix}$$

Then, $\log|\mathbf{G}| = 2\log|\mathbf{A}| + q\log\begin{vmatrix} \sigma_g^2 & \sigma_{gm} \\ \sigma_{gm} & \sigma_m^2 \end{vmatrix} + p\log(\sigma_p^2)$.

The $\log|\mathbf{A}|$ is also a constant but if wanted can be computed easily as a by-product of a one-time calculation of \mathbf{A}^{-1} according to the rules of Quaas (1976) which account for inbreeding. At the end of his algorithm, his vector, \mathbf{v} , corresponds to the square roots of the elements of the diagonal vector of the Choleski factor of \mathbf{A} , i.e., $v_i^2 = \ell_{ii}$.

Thus, since $\mathbf{A} = \mathbf{L}\mathbf{L}'$:

$$\log|\mathbf{A}| = \log|\mathbf{L}| + \log|\mathbf{L}'|, \text{ so that } \log|\mathbf{A}| = 2\sum\log(\ell_{ii}) = 4\sum\log(v_i).$$

Similarly, multiple trait versions of $\log|\mathbf{R}|$ and $\log|\mathbf{G}|$ are not difficult to calculate.

The difficult terms of \mathbf{A} to evaluate are $\log|\mathbf{C}|$ and $\mathbf{y}'\mathbf{P}\mathbf{y}$. The strategy proposed by Smith and Graser (1986) to calculate those terms is based on Gaussian elimination (GE) which is often used to obtain solutions to sets of equations, although in this case solutions are not needed. GE is commonly described in computer science courses. Smith and Graser (1986) took advantage of the steps in GE to calculate $\log|\mathbf{C}|$ as well as $\mathbf{y}'\mathbf{P}\mathbf{y}$. As a simple example suppose the equations are:

$$\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{y} \text{ with } \hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} .$$

Then the residual sum of squares is:

$$(\mathbf{y}'\mathbf{y} - \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y}) = \mathbf{y}'\mathbf{y} - \mathbf{y}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} .$$

As taught in computer science courses for doing GE, Smith and Graser augmented $\mathbf{X}'\mathbf{X}$ as follows with $\mathbf{X}'\mathbf{y}$ and $\mathbf{y}'\mathbf{y}$, the total sum of squares:

$$\begin{array}{c|c} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{y} \\ \hline \mathbf{y}'\mathbf{X} & \mathbf{y}'\mathbf{y} \end{array}$$

Gaussian elimination essentially involves absorbing $\mathbf{X}'\mathbf{X}$ into $\mathbf{y}'\mathbf{y}$ which can be done one equation at a time. The $\log|\mathbf{X}'\mathbf{X}|$ is calculated as the sum of logs of the successive leading diagonals that arise

at each step of the absorption, e.g.,

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = |a_{11}| |a_{22} - a_{21}a_{11}^{-1}a_{12}|, \text{ so that}$$

$$\log \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = \log(a_{11}) + \log(a_{22} - a_{21}a_{11}^{-1}a_{12})$$

After completion of absorption of $\mathbf{X}'\mathbf{X}$ into $\mathbf{y}'\mathbf{y}$, the entry for $\mathbf{y}'\mathbf{y}$ has been replaced by $\mathbf{y}'\mathbf{y} - \mathbf{y}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ which is the residual sum of squares needed for $\mathbf{y}'\mathbf{P}\mathbf{y}$. In this case, $\mathbf{y}'\mathbf{P}\mathbf{y} = [\mathbf{y}'\mathbf{y} - \mathbf{y}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}]/\sigma_e^2$ with σ_e^2 in the denominator because in setting up the OLS the equations were multiplied through by σ_e^2 .

The same procedure works for the general MME:

$$\begin{array}{cc|c} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{Z} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z} + \mathbf{G}^{-1} & \mathbf{Z}'\mathbf{R}^{-1}\mathbf{y} \\ \hline \mathbf{y}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{y}'\mathbf{R}^{-1}\mathbf{Z} & \mathbf{y}'\mathbf{R}^{-1}\mathbf{y} \end{array} = \begin{array}{c|c} \mathbf{C} & \mathbf{r} \\ \hline \mathbf{r}' & \mathbf{y}'\mathbf{R}^{-1}\mathbf{y} \end{array}$$

On completion of absorption of \mathbf{C} , $\log|\mathbf{C}|$ has been computed as the sum of the logs of the successive leading diagonals on absorption and $\mathbf{y}'\mathbf{R}^{-1}\mathbf{y}$ is replaced by $\mathbf{y}'\mathbf{R}^{-1}\mathbf{y} - \mathbf{r}'\mathbf{C}^{-1}\mathbf{r} = \mathbf{y}'\mathbf{P}\mathbf{y}$ which is the generalized residual sum of squares and which does not need to be divided by σ_e^2 because the structure of \mathbf{R}^{-1} is included in the MME, RHS's and total generalized sum of squares, $\mathbf{y}'\mathbf{R}^{-1}\mathbf{y}$.

Karin Meyer (1988,1989,1991) incorporated the ideas of Smith and Graser (1986) into a remarkable series of DFREML programs that took advantage of the sparseness of \mathbf{C} using linked list techniques (e.g., Tier and Smith, 1989). Gaussian elimination is more efficient than inversion to obtain solutions and sparse matrix GE is generally much more efficient in terms of both memory requirements and computing time. The programs include calculation of \mathbf{A}^{-1} with the rules of Quaas (1976). The search strategy for updating \mathbf{R} and \mathbf{G} for the MME was the simplex method of Nelder and Mead (1965) which is a generally efficient method for non-linear optimization and easily accommodates constraining \mathbf{R} and \mathbf{G} to parameter space. Meyer's single-trait programs popularized DFREML and have been widely used. The procedure does not require solutions for $\hat{\boldsymbol{\beta}}$ or $\hat{\mathbf{u}}$ so another program is needed to obtain solutions for $\hat{\boldsymbol{\beta}}$ and $\hat{\mathbf{u}}$ after convergence for estimates of \mathbf{R} and \mathbf{G} ,

probably by iteration.

The DFREML programs expanded by a factor of 5 to 10 the number of equations that could be managed with REML estimation and reduced the computing time for estimation of REML estimates of variance by at least as great a factor. The flatness of the likelihood, might, however, create problems of convergence to local maxima with the search strategies employed with DFREML.

DFREML with Choleski Factorization

Naturally as computers become faster and as more efficient computing algorithms are developed, animal breeders increase the size of analyses they want to do even more. So---what might be more efficient than GE and DFREML? Karin Meyer has continued to make her programs more efficient and more general. Misztal (1990) showed that sparse matrix techniques could reduce time and memory requirements for EM-REML algorithms. He also suggested use of such techniques for DFREML computation. Boldman and Van Vleck (1991), based on the suggestion of Misztal, worked at trying to incorporate sparse matrix routines other than GE into an algorithm to calculate Λ . Their strategy was based on Choleski factorization and solves using sparse matrix routines in SPARSPAK (George et al., 1980; George and Ng, 1984; Chu et al., 1984). Choleski factorization is a matrix technique involved in many advances in animal breeding computing, e.g., the rules for \mathbf{A}^{-1} , simulation of multiple trait data sets, canonical transformation for certain multiple trait analyses (e.g., Meyer, 1985) and sequential transformation to standardize residual variances for one class of multiple trait analyses with chronological selection on the traits (e.g., Pollak and Quaas, 1982; Walter et al., 1986).

The Choleski factor, \mathbf{L} , for a symmetric positive definite matrix, \mathbf{C} , is such that $\mathbf{LL}' = \mathbf{C}$ with \mathbf{L} , a lower triangular matrix. For example:

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{12} & c_{22} & c_{23} \\ c_{13} & c_{23} & c_{33} \end{bmatrix}$$

\mathbf{L} can be calculated easily from the obvious (after you have done it once) recursive pattern.

For MME of the form: $\mathbf{Cs} = \mathbf{r}$,

the two difficult terms of Λ , $\log|\mathbf{C}|$ and $\mathbf{y}'\mathbf{P}\mathbf{y}$, can be calculated easily and, with sparse matrix

techniques, very rapidly for many animal breeding analyses.

First, find \mathbf{L} such that $\mathbf{LL}' = \mathbf{C}$, then $\mathbf{LL}'\mathbf{s} = \mathbf{r}$.

Let $\mathbf{d} = \mathbf{L}'\mathbf{s}$ so that $\mathbf{Ld} = \mathbf{r}$.

The down-solve to calculate the vector \mathbf{d} is obviously easy.

Next up-solve $\mathbf{L}'\mathbf{s} = \mathbf{d}$ for \mathbf{s} which is just as easy.

From \mathbf{s} , $\mathbf{y}'\mathbf{P}\mathbf{y}$ can be calculated as $\mathbf{y}'\mathbf{R}^{-1}\mathbf{y} - \mathbf{s}'\mathbf{r}$.

From the relationship $\mathbf{C} = \mathbf{LL}'$, $\log|\mathbf{C}| = \log|\mathbf{L}| + \log|\mathbf{L}'|$.

The log determinant of a lower triangular matrix is simply $\sum \log(\ell_{ii})$ where the ℓ_{ii} are the diagonals of \mathbf{L} . Also obvious is that $\log|\mathbf{L}| = \log|\mathbf{L}'|$. Thus $\log|\mathbf{C}| = 2 \sum [\log(\ell_{ii})]$ which is a trivial step. The other steps in the Boldman and Van Vleck (1991) algorithm are the same as with the original DFREML programs and make use of the Simplex routine in updating \mathbf{R} and \mathbf{G} to maximize Λ .

The decrease in computing time with the SPARSPAK-Choleski strategy over the original GE with linked list was nearly unbelievable. For a problem with 3661 equations, the time to calculate the likelihood decreased from 462.7 to .5 sec on an IBM 3090 supercomputer, from 2591.4 to 3.3 sec on an IBM 4381 mainframe, and from 3271.2 to 12.0 sec on a 386/20 personal computer--decreases in computing time of 200 to 900 times. The SPARSPAK algorithm, however, requires a one-time reordering for each design matrix which for the example took 28.7, 153.3, and 243.9 sec for the three computers. The number of likelihoods evaluated to obtain convergence for the example was 157 and is often more for other data sets and models. The Choleski factorization is not more efficient than Gaussian elimination. The advantage with SPARSPAK is in reordering but SPARSPAK does not have the GE option. K. Meyer (1991, personal communication) by obvious reordering found GE to be nearly as fast as SPARSPAK in newer versions of her programs.

Other packages for sparse matrix routines are SMPAK and MATLAB, an interactive, programmable matrix package often used for examples in teaching. The key to the efficiency with SPARSPAK is that a particular data structure needs to be reordered only once to minimize steps that occur each round in Choleski factorization. The package remembers the reordering when the coefficients of \mathbf{C} and the RHS's, updated for new guesses of \mathbf{R} and \mathbf{G} , are entered each round.

The SPARSPAK factorization requires that the coefficient matrix be of full rank; i.e., that constraints be imposed on the coefficient matrix. In some cases determining the proper constraints is difficult and in most cases is a decided nuisance. Kachman (Chapter 6) modified the Choleski factorization of SPARSPAK so that zero-out type constraints are imposed automatically during the

factorization step.

The general approach is easily adapted to multiple trait problems and seems to be the most efficient procedure available, at least until a better one is developed. A simple two trait analysis with one trait measured only on males and the other trait measured only on females with 52,192 equations took about 9 and 13 minutes per likelihood evaluation on personal computers with Intel 486/50-25 and Intel 486/33 processors. For EM-REML, the matrix of order 52,000 would need to be inverted each round, an impossibility with standard inversion programs even with the fastest super computers. The symmetric half-stored form of the matrix would also require 11,000,000,000 bytes of memory!! Therefore DFREML-SPARSPAK can do the impossible in 9 to 13 minutes on a PC and with a little over 16,000,000 bytes of memory. Misztal (1990) with sparse matrix inversion inverted the coefficient matrix for a single trait model of order 74,199 once in 5 hr using 64MB of memory on a CRAY supercomputer and then sampled elements of the inverse in succeeding rounds.

The procedure outlined here is completely general--single traits, multiple traits, direct genetic effects, maternal genetic effects, permanent environmental effects, litter effects, numerator relationships, dominance relationships, and cytoplasmic effects. Multiple traits can have different fixed factors, traits can be missing, traits can have repeated measures, and sex-limited traits of both sexes can be included. The most obvious problem with such a general procedure with possibly many parameters in \mathbf{R} and \mathbf{G} even for only 3-4 traits, is whether convergence to estimates of \mathbf{R} and \mathbf{G} that maximize Λ will occur. At the least, MTDFREML programs should be restarted with what are considered converged estimates to determine whether a local maximum was found (Groeneveld and Kovac, 1990). A larger Λ on restart indicates the original maximum was local. Experience and, perhaps, luck may be necessary to obtain the global maximum for Λ . A cold restart that yields a larger log likelihood in the units or tenths position probably indicates the previous convergence was to a local maximum. A particular problem with small data sets, which may be a function of poor design matrices for separation of direct and maternal genetic effects, is a tendency for the genetic correlation to converge towards 1 or -1.

Hypothesis Tests and Prediction Error Variance

The Choleski based algorithm lends itself to calculation of standard errors for solutions, for standard errors of linear (estimable) functions of solutions, and for prediction error variances for

random effects such as breeding values. Harville (1979) presented what he called the mixed model conjugate normal equations based on a suggestion in Henderson (1974). Beneath the complicated name was a method waiting to exploit the rapid solution time with the sparse Choleski factorization. The algebra of the method is simple. Let $\mathbf{s}' = (\hat{\boldsymbol{\beta}}' \hat{\mathbf{u}}')$ be the solutions at convergence for \mathbf{R} and \mathbf{G} . Then from mixed model theory under the pretense that \mathbf{R} and \mathbf{G} are known exactly:

$$\text{Var} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{u}} \end{bmatrix} = \mathbf{C}^{-1}$$

where \mathbf{C} is the full rank coefficient matrix for the MME. To test the significance of an estimable function of the $\hat{\boldsymbol{\beta}}$'s, e.g., $\mathbf{k}'(\hat{\boldsymbol{\beta}}' \hat{\mathbf{u}}' - \mathbf{u}')$ the variance of $(\hat{\boldsymbol{\beta}}' \hat{\mathbf{u}}' - \mathbf{u}')$ is needed, i.e., $\mathbf{k}'\mathbf{C}^{-1}\mathbf{k}$. The question is how to obtain $\mathbf{k}'\mathbf{C}^{-1}\mathbf{k}$ without knowing \mathbf{C}^{-1} ?

The approach is to use the contrast vector, \mathbf{k} , as the RHS vector of the MME and let Φ be the solution:

$$\mathbf{C}\Phi = \mathbf{k}, \text{ so that algebraically } \Phi = \mathbf{C}^{-1}\mathbf{k}.$$

Then premultiplying Φ by \mathbf{k}' gives $\mathbf{k}'\Phi = \mathbf{k}'\mathbf{C}^{-1}\mathbf{k}$ which is what is needed. Rather than use \mathbf{C}^{-1} in calculation of Φ , the Choleski factor of \mathbf{C} , \mathbf{L} , is used to obtain Φ just as \mathbf{s} was obtained (and just as quickly and with the same reordered structure, see Table 1). For the i th linear contrast, \mathbf{k}_i 's, the standard error is $(\mathbf{k}_i'\Phi_i)^{.5}$ which can be used for t-tests or the $\mathbf{k}_i'\Phi_i$ for the orthogonal contrasts within a factor can be summed to obtain a F-test.

As an example, assume three levels of the first factor, then:

$$\mathbf{k} = (1 \ 0 \ -1 \ 0 \ 0 \ \dots \ 0)'$$

is the contrast to test level 1 minus level 3 of factor one.

Prediction error variances can be obtained similarly. In such cases, \mathbf{k} will contain all zeroes except for a 1 in the j th spot corresponding to the equation for the j th breeding value. Again solve $\mathbf{L}\mathbf{L}'\Phi_j = \mathbf{k}_j$ for Φ_j and then calculate $V(\hat{u}_j - u_j)$ as $\mathbf{k}_j'\Phi_j$ which can be used to obtain accuracy of prediction or to put a confidence range about u_j .

Again, the reminder. The mixed model conjugate normal equation procedure is general--single, multiple traits, messy, balanced data, etc.

Multiple Trait MME for DFREML

The basic computing strategy for calculating the general MME is fairly simple. The problem is in creating enough flexibility to handle missing records, multiple traits, multiple genetic and other random factors, different fixed factors for different traits, repeated measures as well as incorporating

relationship matrices especially for multiple genetic effects such as direct and maternal genetic effects.

Expectations of Solutions

The Choleski algorithm also leads to relatively easy determination of the expected values of solutions. Even with constraints imposed explicitly, expectations of solutions are not always obvious. With the constraints imposed by the Kachman modifications of SPARSPAK, knowledge of expected values of the solutions becomes even more important. A property of MME is that predictions of \mathbf{u} have expected values of zero. Thus, verification that expected values of $\hat{\mathbf{u}}$ are zero may be an aid in debugging.

Algebra of Expected Values

With $\mathbf{C}^{-} = \begin{bmatrix} \mathbf{C}^{\text{XX}} & \mathbf{C}^{\text{XZ}} \\ \mathbf{C}^{\text{ZX}} & \mathbf{C}^{\text{ZZ+}} \end{bmatrix}$, the generalized inverse of the coefficient matrix, $\mathbf{s} = (\boldsymbol{\beta}' \hat{\mathbf{u}})'$, the solution vector, and $\mathbf{r} = \begin{bmatrix} \mathbf{X}' \mathbf{R}^{-1} \mathbf{y} \\ \mathbf{Z}' \mathbf{R}^{-1} \mathbf{y} \end{bmatrix}$, the right hand side vector:

$$\mathbf{s} = \mathbf{C}^{-} \mathbf{r} \text{ and } E[\mathbf{s}] = E[\mathbf{C}^{-} \mathbf{r}] = \mathbf{C}^{-} E[\mathbf{r}],$$

then

$$E[\mathbf{s}] = \begin{bmatrix} \mathbf{C}^{\text{XX}} & \mathbf{C}^{\text{XZ}} \\ \mathbf{C}^{\text{ZX}} & \mathbf{C}^{\text{ZZ+}} \end{bmatrix} \begin{bmatrix} \mathbf{X}' \mathbf{R}^{-1} \mathbf{y} \\ \mathbf{Z}' \mathbf{R}^{-1} \mathbf{y} \end{bmatrix} \quad E[\mathbf{y}] = \begin{bmatrix} \mathbf{C}^{\text{XX}} & \mathbf{C}^{\text{XZ}} \\ \mathbf{C}^{\text{ZX}} & \mathbf{C}^{\text{ZZ+}} \end{bmatrix} \begin{bmatrix} \mathbf{X}' \mathbf{R}^{-1} \mathbf{X} \\ \mathbf{Z}' \mathbf{R}^{-1} \mathbf{X} \end{bmatrix} \boldsymbol{\beta}.$$

Let \mathbf{c}'_i be the i^{th} row of \mathbf{C}^{-} .

Then

$$E[\mathbf{S}] = \begin{bmatrix} \mathbf{c}'_1 \\ \mathbf{c}'_2 \\ \vdots \\ \mathbf{c}'_i \end{bmatrix} \begin{bmatrix} \mathbf{X}' \mathbf{R}^{-1} \mathbf{y} \\ \mathbf{Z}' \mathbf{R}^{-1} \mathbf{y} \end{bmatrix} \boldsymbol{\beta}.$$

Note that the expectation of the i^{th} solution in \mathbf{s} :

$$E[S_i] = \mathbf{c}_i' \begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{y} \end{bmatrix} \boldsymbol{\beta}.$$

Now note that $\begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{X} \end{bmatrix}$ consists of the first \mathbf{p} columns in \mathbf{C} or the transpose of the part of the coefficient matrix associated with the $\boldsymbol{\beta}$ equations. Thus, multiplication of the i^{th} row of \mathbf{C} by each of the first \mathbf{p} columns of \mathbf{C} gives the coefficients of $\boldsymbol{\beta}$ in $E[s_i]$. Note that \mathbf{c}_i can be obtained by solving $\mathbf{C}\mathbf{c}_i = \mathbf{I}_i$ where \mathbf{I}_i is the i^{th} column of \mathbf{I} (a 1 in the i^{th} element and zeroes elsewhere). As with determining \mathbf{s} the solution vector, \mathbf{c}_i can be determined by a Choleski solve from $\mathbf{L}\mathbf{L}'\mathbf{c}_i = \mathbf{I}_i$. To calculate coefficients of $\boldsymbol{\beta}$ in $E[s_i]$, \mathbf{c}_i is multiplied by non-zero coefficients of the first \mathbf{p} columns of \mathbf{C} while reading the least squares coefficients as is done in setting up the MME. This process is repeated for each expected value. Because the coefficient matrix needs to be factored only once, the costs for each expected value are: 1) to change the RHS to insert zeroes and a 1 corresponding to solution i , 2) to use a Choleski solve to find \mathbf{c}_i , and 3) to do sparse multiplication of \mathbf{c}_i by non-zero coefficients of \mathbf{C} corresponding to the $\boldsymbol{\beta}$ equations. Reading the coefficients of \mathbf{C} is most costly but is not prohibitive for sets of solutions such as for all levels of certain fixed factors or for a few levels of random factors for verification of the zero expected values. (See Table 1 for an example of the time required for one expected value).

Table 1. Relative computational times (seconds on a 486/33) for various options: iterate for variance components, solution to mixed model equations, calculation of standard errors and standard deviations of prediction errors, and calculation of expected values for a single trait analysis with 3,111 animals and 7,303 equations

Variance Components (Data dependent)		Solutions MME (Data & VC dependent)	
1) Reorder	(98.20)		
2) Factor	(44.60)	2) Factor	(44.60)
3) Solve	(1.32)	3) Solve	(1.32)
4) Repeat 2) & 3) until convergence	(45.92)		
Standard Errors (Data & VC dependent)		Expected Values (Design dependent)	
2) Factor	(44.60)	2) Follows SE	
3) Solve	(1.32)	3) Solve	(1.32)

4) Repeat 3) for each contrast (1.32)

4) Mult. $X'R^{-1}X$, $Z'R^{-1}X$ (4.01)

5) Repeat 3 & 4) (5.33)
for each $E[]$

CHAPTER FIVE: Computing Strategies for MTDFREML

Estimation of variance components via MTDFREML consists of two distinct and independent steps: 1) an orderly method to update \mathbf{G} and \mathbf{R} in an attempt to locate the values which maximize the log likelihood for the sample of data, Λ (equivalently, minimize, $FVALUE = -2\Lambda$), and 2) formation and solution of the MME to evaluate -2Λ for specific values of \mathbf{G} and \mathbf{R} . The derivative-free search procedure used in MTDFREML is the Simplex method of Nelder and Mead (1965) described in Chapter 7. The procedure used for formation and solution of the MME will be described in this section.

The use of a sparse matrix package such as SPARSPAK which uses a Choleski factorization greatly reduces both computing time and amount of programming required to obtain MTDFREML estimates. Several characteristics of SPARSPAK make the routines especially well-suited for forming and solving the MME in MTDFREML. First, individual elements of the MME can be entered in any order and are accumulated by SPARSPAK. Thus, elements of $\mathbf{X}'\mathbf{R}^{-1}\mathbf{X}$, $\mathbf{Z}'\mathbf{R}^{-1}\mathbf{X}$, and $\mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z}$ and $\mathbf{X}'\mathbf{R}^{-1}\mathbf{y}$ and $\mathbf{Z}'\mathbf{R}^{-1}\mathbf{y}$ can be accumulated one animal at a time. After all records have been processed, elements of \mathbf{G}^{-1} can then be added individually to elements of $\mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z}$. Second, SPARSPAK stores diagonal elements of the Choleski factor, \mathbf{L} , of the coefficient matrix, and the right-hand side vector, \mathbf{r} , and solution vector, \mathbf{s} , in separate vectors so the quantities $\sum \log(l_{ii})$ and $\mathbf{s}'\mathbf{r}$ required to evaluate Λ in each round can be easily obtained. Third, (see Chapter 6), Choleski factorization of SPARSPAK can be easily modified to impose zero-type constraints automatically during the factorization step so that the user need not define particular constraints, although the option to impose specific constraints is available. In addition, the reordering, factorization, and solution steps performed by SPARSPAK are invisible to the user so that knowledge of the details of reordering algorithms or techniques for storing and manipulating sparse matrices is not required. The MTDFREML programs simply provide the non-zero elements of the lower-half of \mathbf{C} , the MME coefficient matrix, and \mathbf{r} , the MME right-hand sides. SPARSPAK then reorders the system (but once only) and obtains for each evaluation of Λ , the Choleski factor and solution vector \mathbf{s} , which are then used to calculate $\log |\mathbf{C}|$ and $\mathbf{s}'\mathbf{r}$.

MTDFREML consists of three programs: MTDNFNM which forms the non-zero elements of \mathbf{A}^{-1} ; MTDFPREP which forms the non-zero elements of $(\mathbf{X} \ \mathbf{Z})=\mathbf{W}$ for each animal; and MTDFRUN which updates \mathbf{G} and \mathbf{R} , forms and solves the MME and calculates Λ in each round of iteration. Because only minor modifications have been made to the original version of DFNRM written by Karin Meyer (1988), following the method of Quaas (1976), the strategy used in that

program will not be described. In contrast, MTDFPREP and MTDFRUN were written specifically for MTDFREML so the strategies used in each will be explained to facilitate modifications for analyses not currently supported. Basically, MTDFPREP forms the part of the MME which is independent of the values of \mathbf{G} and \mathbf{R} used in each round, i.e., $\mathbf{W}=(\mathbf{X} \mathbf{Z})$ and \mathbf{y} . In addition, the program determines the non-zero structure of \mathbf{R} for each animal and the structure of \mathbf{G}^{-1} to be added to the MME. During each round of iteration in MTDFRUN, \mathbf{R} and \mathbf{G} are updated. Then \mathbf{W}_i and \mathbf{y}_i are read for each animal i and the non-zero elements of the weighted least-squares equations are generated by forming $\mathbf{W}_i' \mathbf{R}_i^{-1} \mathbf{W}_i$ and $\mathbf{W}_i' \mathbf{R}_i^{-1} \mathbf{y}_i$. The weighted sum of squares is accumulated from $\mathbf{y}_i' \mathbf{R}_i^{-1} \mathbf{y}_i$. Next, the non-zero elements of \mathbf{G}^{-1} are added to the weighted least-squares equations to form the MME. Finally, SPARSPAK is used to solve the MME and obtain the terms required to calculate $-2\mathbf{A}$. MTDFREML can accommodate a wide range of models (see Chapters 2 and 3).

MTDFPREP

With appropriate modification of the include file (PARAM.DAT), this program can fit any number of fixed effects (both discrete and continuous) and several random effects in addition to the required animal effect. Models can be different for each trait and missing observations are permitted. Repeated records can be fit but the possible variance structures are limited in the current version. Other options will likely be added as the need arises.

MTDFPREP reads the data from unit 33 (ascii) which is set up with integer variables followed by real variables. The data file is read twice, first to determine the number of levels for each discrete factor and the simple statistics for each continuous variable (covariates and traits), and second to recode levels of factors to correspond to the order of the MME and to express each continuous variable as a deviation from its mean.

To illustrate the strategy used in the programs, data from Meyer (1991) will be used. Records are body weight (t1) and intake (t2) measured on 284 animals. The pedigree file includes 55 base animals for a total of 339 animals in \mathbf{A} . For each trait the model of analysis includes random animal (a) and uncorrelated litter (lit: 42 levels) effects. Fixed effects are litter size (lsc: covariate) and generations (gen: 3 levels) for body weight, and litter size (lsd: 7 levels) and sex (sex: 2 levels) for intake. The first two records in the data file (7 integers and 3 reals) are:

animal	sire	dam	gen	sex	lsd	lit	lsc	t1	t2
20101	11012	10101	1	1	4	1	4.0	22.5	59.1
20102	11012	10101	1	1	4	1	4.0	22.6	0.0

Note that litter size appears twice in the data, both as the sixth integer (lsc) and as the first real (lsc) variable. In addition, intake was missing for animal 20102 so the field for t2 is coded as 0.0 which is used as the missing value.

The first step is to run MTDFNRM which forms A^{-1} and writes the sorted vector of 329 animal IDs to unit 11 (ascii) and the lower-half stored non-zero elements to unit 44 (binary). MTDFPREP is then run with the following parameters in batch file, MTDFP5.DAT (see Chapter 2):

MUEX2UR.DAT	name of data file	
test of MTDFPREP	with km mouse data	
2	traits	
*	end of comments	
7	# integers	
3	# reals	
2	# traits	
weight	trait name	trait 1
2	trait position in reals	
0.0	value for missing	
1	# covariates	
litter size	name cov. 1	
1	cov. 1 position in reals	
1	order for cov. 1	
1	# fixed effects	
generation	name fixed 1	
4	fixed 1 position in integers	
0	write levels to unit 66: 1 yes; 0 no	
1	animal position in integers	
339	animals in A-1	
0	2nd animal	
1	uncorrelated random	
litter	name uncorrelated 1	
7	uncorr. 1 position in integers	
0	write levels to unit 66: 1 yes; 0 no	
intake	trait name	trait 2
3	trait position in reals	
0.0	value for missing	
0	# covariates	
2	# fixed effects	
litter size	name fixed 1	
6	fixed 1 position in integers	
0	write levels to unit 66: 1 yes; 0 no	
sex	name fixed 2	
5	fixed 2 position in integers	
0	write levels to unit 66: 1 yes; 0 no	
0	2nd animal	
1	uncorrelated random	
litter	name uncorrelated 1	
7	uncorr. 1 position in integers	
0	write levels to unit 66: 1 yes; 0 no	

The subsequent steps are:

1. Read the number of animal IDs from unit 11 and compare to the number entered by the user as read from the log of MTDENRM (e.g., 339). If these numbers are not equal, the program terminates because the wrong pedigree file is probably being used. If the numbers are equal, the sorted vector of IDs is read from unit 11.

First read of data:

2. The 284 lines of data are then read sequentially from unit 33. For each line, all (7) integer variables are read into an integer vector and all (3) real variables are read into a real vector. Each of the $j = 2$ traits is then processed:
 - a) If the value for the trait is equal to the missing value (0.0), skip to the next trait.
 - b) If the value for the trait is valid:
 - i) increment the count, sum, and sum of squares for each real variable,
 - ii) compare the value of each fixed factor and uncorrelated random factor (e.g., litters) to the unique numeric (but unsorted) list of current values stored in memory; if the value is not already in the list, it is added at the end and the number of levels for the effect is incremented by one.
3. After all lines of data have been read into memory, sort the vectors of levels for each discrete fixed and uncorrelated random factor.
4. Calculate the mean and variance for each real variable.
5. Based on the sequence of the MME and the number of levels for each factor, determine the starting row of each factor in the model. The starting row is expressed as one less than the actual position. For the example data the structure is:

Factors	(starting row MME)-1	No. rows
t1: linear covariate for litter	0	1
t1: fixed effect for	1	3
t2: fixed effect for litter size	4	7
t2: fixed effect for sex	11	2
t1: random animal	13	339
t2: random animal	352	339
t1: random litter	691	42
t2: random litter	733	42

For this example, there are 775 equations in the MME. The means for trait 1 are 4.4789 for the litter size covariate and 24.0687 for weight, and for trait 2 the mean for intake is 64.2975.

Second read of data:

6. The 284 lines of data are then reread sequentially from unit 33. For each line, all (7) integer variables are read into an integer vector and all (3) real variables are read into a real vector. Each

of the $j = 2$ traits is then processed:

- a) If the value for the trait is equal to the missing value (0.0), skip to the next trait.
- b) If the value for the trait is valid, the value and position of each value in \mathbf{W} and \mathbf{y} is then determined:
 - i) covariates and observations are deviated from their corresponding means, e.g., the deviations for the first record are: $4.0-4.4789=-0.4789$ (litter size), $22.5-24.0687=-1.5687$ (weight), and $59.1-64.2975=-5.1975$ (intake).
 - ii) the \mathbf{W} row position of each regression coefficient is determined from the sequence and order (linear, quadratic, etc.) of the covariates,
 - iii) the position of each discrete factor in \mathbf{W} is determined by looking up its position in the corresponding vector of sorted levels and then adding this position number to the starting row position for the factor; e.g., for the first record, the value of 4 for litter size (trait 2) is found at position 4 in the sorted list of levels (1, 2,..., 6, 7) so 4 is added to the starting position (4) for litter size to give the row position in \mathbf{W} of 8.

7. After each line of data is read, the values and positions of elements in \mathbf{W}_{ij} and the value for \mathbf{y}_{ij} are written to unit 51 (binary); the length of \mathbf{W}_{ij} is determined by the total number of model effects and the number of valid traits, and the number of \mathbf{W}_{ij} rows is equal to the number of valid traits.
8. After all lines of \mathbf{W}_{ij} have been written to unit 51 for a record, the column positions are written to unit 51 and a summary for each animal is written to unit 52; this information consists of animal number and number of data lines, effects, traits (rows in \mathbf{W}_{ij}), and structure (i.e., pattern of missing values) of observations for the animal.

For the record of animal one, the values written to unit 51 and 52 are (subscript denotes trait number; text in parentheses is not written):

unit 51 - values and positions for \mathbf{W}_{ij} :

	(lsc ₁)	(gen ₁)	(lsc ₂)	(sex ₂)	(a ₁)	(a ₂)	(lit ₁)	(lit ₂)	(y _i)
(t ₁)	-0.479	1.0	0.0	0.0	1.0	0.0	1.0	0.0	-1.5687
(t ₂)	0.0	0.0	1.0	1.0	0.0	1.0	0.0	1.0	-5.1975
(row)	1	2	8	12	69	408	692	734	

unit 52 - summary of information for first animal (two lines):

(Anim no.)	(No. data lines in unit 33 for the animal)		(Trait structure)	
1	1		(p ₁)	(p ₂)
(No. effects)	(No traits)	(code)		
8	2	3	1	1

Trait structure values are used to indicate the form of \mathbf{R}_i^{-1} to be used in $\mathbf{W}_i' \mathbf{R}_i^{-1} \mathbf{W}_i$ and $\mathbf{W}_i' \mathbf{R}_i^{-1} \mathbf{y}_i$, and consist of a value (p_j) for each trait of 1 or 0, if the trait is present or missing, respectively, and a code calculated as $\sum_j (p_j * 2)^{(j-1)}$.

In the record of the second animal, trait 2 (intake) is missing so a single \mathbf{W}_{ij} row of four effects is written to unit 51:

	(lsc ₁)	(gen ₁)	(a ₁)	(lit ₁)	(y ₁)
(t ₁ :)	-0.4789	1.0	1.0	1.0	-1.4687
(row)	1	2	70	692	

The information written to unit 52 is (two lines):

(Anim no.)	(No. data lines)		(Trait structure)	
2	1		(p ₁)	(p ₂)
(No. effects)	(No traits)	(code)		
4	1	1	1	0

- After all lines of data have been read, information describing the models and MME (e.g., number of traits, starting position and number of levels for each effect) is written to unit 50 (ascii) and trait names and original identification for levels of factors to unit 67 (ascii). A summary of the model and data is then written to unit 66 (MTDF66 in ascii); unlike other files written in MTDFPREP, this file is not read by MTDFRUN.

MTDFRUN

In this program, the information on \mathbf{W}_i and structure of \mathbf{R}_i for each animal is read and the MME are formed for the current values of \mathbf{G} and \mathbf{R} . In the main section of MTDFRUN, the starting values of \mathbf{G} and \mathbf{R} are input (first round) and the initial Simplex is setup and updated in each subsequent round. The current values of \mathbf{G} and \mathbf{R} are passed to the subroutine MTDFLIK which sets up and solves the MME via SPARSPAK and returns the value of $-2\Lambda = \text{FVALUE}$. (The constant, $N \log(2\pi)$ is not included.)

For the first run, the sequential steps in MTDFRUN are (also see Chapter 2):

1. Information on the models and MME is read from unit 50 (ascii). The program will query for user-designated constraints (optional) to make the MME full rank and then ask if the MME have already been reordered. Based on the model information, the program will query for starting values for the covariance matrices in the order animal (required) and second animal (optional), uncorrelated random (optional), and residual effects (required). The option to hold any starting value constant is provided. Then the matrices of starting values are transformed to a single vector to be passed to subroutine MTDFLIK. Convergence criterion and maximum number of Simplex rounds are also input.
2. The value of -2Λ for the starting values of the covariance components is calculated in MTDFLIK. These starting values correspond to the first vertex of the initial Simplex. If any of the starting values are non-permissible, e.g., h^2 greater than 1, the likelihood is not evaluated and the program is stopped. If the initial point is permissible, an additional Simplex vertex is evaluated for each (co)variance component that is not held constant. The additional points are obtained by multiplying each (co)variance component in turn by 1.2, i.e., step size of 0.2. If the parameter values for any of the original vertices are non-permissible, the component is multiplied by .7 until it is permissible. After the points of the initial Simplex are evaluated, in each round the worst point of the Simplex (largest -2Λ value) is replaced as described in Chapter 7 until convergence is attained or the maximum number of rounds is reached.

In each round of iteration, the (co)variance priors for which to evaluate -2Λ are determined in MTDFRUN and passed to subroutine MTDFLIK to be evaluated. The sequential steps in subroutine MTDFLIK are:

1. Information on model and MME is read from unit 50 (ascii). The phenotypic variance is calculated for each trait. Values of all parameters (e.g., h^2 , c^2 , r_c , r_g , r_e , and r_p) are checked for permissibility. If all are valid, each of $2^t - 1$ possible forms of \mathbf{R}_i is set-up, where r is the order of the full \mathbf{R}_i matrix for an animal, i.e., number of traits. The inverse of each \mathbf{R}_i matrix is obtained, along with the inverses of covariance matrix of genetic animal effects, \mathbf{G}_o and covariance matrix

of uncorrelated random effects, \mathbf{C}_o . Before inverting these matrices, the eigenvalues are calculated as an additional permissibility check. If any parameter is invalid or any eigenvalue is negative, the likelihood is not evaluated but instead a very large likelihood value (1.e+37) is assigned to the set of priors which will force a contraction in the Simplex. The sums of the logs of the eigenvalues are also used to calculate $\log|\mathbf{G}_o|$ and $\log|\mathbf{R}_i|$. If all priors are valid, the corresponding MME are set up one animal at a time using SPARSPAK. The reordering performed by SPARSPAK is dependent on the location of non-zero elements in MME (which do not change over rounds of iteration). Therefore, in the first round of iteration when the MME have not yet been reordered, only the positions of the non-zero elements are input to SPARSPAK; in later rounds when the reordering is known both positions and values of the non-zero elements are input to SPARSPAK.

a) The two lines of summary information for each animal are read from unit 52: line 1 for animal number and number of data lines; and line 2 for number of effects, traits (rows in \mathbf{W}_{ij}), and structure (i.e., pattern of missing values) of observations for the animal. The structure of observations determines the appropriate form of \mathbf{R}_i to be used in $\mathbf{W}'_i\mathbf{R}_i^{-1}\mathbf{W}_i$ and $\mathbf{W}'_i\mathbf{R}_i^{-1}\mathbf{y}_i$ and the contribution to $\log|\mathbf{R}_i|$.

b) The values of \mathbf{W}_i and \mathbf{y}_i for each animal are then read from unit 51 into a matrix and after all rows of \mathbf{W}_i for an animal are input, the corresponding column positions are read from unit 51. For the example data, the $\mathbf{W}_i | \mathbf{y}_i$ values for the first animal (two records) are:

$$\begin{array}{cccccccc|c} -.04789 & 1.0 & 0.0 & 0.0 & 1.0 & 0.0 & 1.0 & 0.0 & -1.5687 \\ 0.0 & 0.0 & 1.0 & 1.0 & 0.0 & 1.0 & 0.0 & 1.0 & -5.1975 \end{array}$$

and the column positions read for \mathbf{W}_i are:

$$1 \quad 2 \quad 8 \quad 12 \quad 69 \quad 408 \quad 692 \quad 734$$

c) The correct form of \mathbf{R}_i^{-1} is then used to form $\mathbf{W}'_i\mathbf{R}_i^{-1}\mathbf{W}_i$ and $\mathbf{W}'_i\mathbf{R}_i^{-1}\mathbf{y}_i$ for each animal. Let the prior values for the full \mathbf{R}_i be:

$$\begin{pmatrix} 1.7 & 1.0 \\ 1.0 & 12.6 \end{pmatrix}$$

The full form of \mathbf{R}_i^{-1} for animals with records for both traits is:

$$\begin{pmatrix} .617 & -.049 \\ -.049 & .083 \end{pmatrix}$$

For the first animal with both traits recorded, $\mathbf{W}'_i\mathbf{R}_i^{-1}\mathbf{W}_i$ is (row and column positions in parentheses):

	(1)	(2)	(8)	(12)	(69)	(408)	(692)	(734)
(1)	.1415	-2955	.0235	.0235	-.2955	.0235	-.2955	.0235
(2)	-2955	.6170	-.0490	-.0490	.6170	-.0490	.6170	-.0490
(8)	.0235	-.0490	.0833	.0833	-.0490	.0833	-.0490	.0833
(12)	.0235	-.0490	.0833	.0833	-.0490	.0833	-.0490	.0833
(69)	-.2955	.6170	-.0490	-.0490	.6170	-.0490	.6170	-.0490
(408)	.0235	-.0490	.0833	.0833	-.0490	.0833	-.0490	-.0833
(692)	-.2955	.6170	-.0490	-.0490	.6170	-.0490	.6170	-.0490
(734)	.0235	-.0490	.0833	.0833	-.0490	.0833	-.0490	.0833

and $\mathbf{W}_i' \mathbf{R}_i^{-1} \mathbf{y}_i$ is:

(1)	-.5854
(2)	1.2225
(8)	-.5095
(12)	-.5095
(69)	1.2225
(408)	-.5095
(692)	1.2225
(734)	-.5095

These non-zero elements are then input for SPARSPAK to accumulate, e.g., .1415 corresponds to position (1,1) of the left-hand side of the MME and -.5854 corresponds to row one of the right-hand side. $\mathbf{W}_i' \mathbf{R}_i^{-1} \mathbf{y}_i$ elements are summed and stored in a vector and then input into SPARSPAK after all records are processed.

For the second animal with only trait one recorded, with covariate = -.4780, record = -1.4687, and $\mathbf{R}_i^{-1} = 1/1.7 = .5882$, $\mathbf{W}_i' \mathbf{R}_i^{-1} \mathbf{W}_i$ is:

	(1)	(2)	(70)	(692)
(1)	.1349	-.2817	-.2817	-.2817
(2)	-.2817	.5882	.5882	.5882
(70)	-.2817	.5882	.5882	.5882
(692)	-.2817	.5882	.5882	.5882

and $\mathbf{W}_i' \mathbf{R}_i^{-1} \mathbf{y}_i$ is:

(1)	.4137
(2)	-.8639
(70)	-.8639
(692)	-.8639

d) Along with $\mathbf{W}_i' \mathbf{R}_i^{-1} \mathbf{W}_i$ and $\mathbf{W}_i' \mathbf{R}_i^{-1} \mathbf{y}_i$, the contribution of each animal to the scalar $\mathbf{y}_i' \mathbf{R}_i^{-1} \mathbf{y}_i$ is also accumulated.

2. After the non-zero elements of the weighted least-squares portion of the MME for records of all animals have been processed and accumulated in SPARSPAK, the elements of \mathbf{G}^{-1} are then added to form the complete MME.
 - a) Each non-zero element \mathbf{a}^{ij} (and its row and column position) of lower-half stored \mathbf{A}^{-1} is then read from unit 44 (binary) and multiplied by the appropriate elements of the current \mathbf{G}_0^{-1} . For the example data with two traits and no additional correlated random effect, each diagonal element of \mathbf{A}^{-1} contributes three values to the MME, i.e., \mathbf{a}^{ii_g11} , \mathbf{a}^{ii_g12} , and \mathbf{a}^{ii_g22} and each off-diagonal element of \mathbf{A}^{-1} contributes four values to the MME (the covariance section is full-stored so the transpose of \mathbf{a}^{ij_g12} is also added). Row and column positions are determined by adding to the row and column positions of \mathbf{a}^{ij} , the starting row for each animal effect.
 - b) Elements of the inverse of the covariance matrix of uncorrelated random factors corresponding to uncorrelated random effects are then added to the MME. These contributions consist of the inverse elements multiplied by an identity matrix of order equal to the number of levels of the effect. For the example data with the same uncorrelated random factor (litter) for each trait, these contributions are \mathbf{I}^*_{c11} , \mathbf{I}^*_{c12} , and \mathbf{I}^*_{c22} , where \mathbf{I} is of order 42.
3. After all non-zero elements of the MME have been passed to SPARSPAK, the MME equations are then either reordered (initial round of iteration) or solved (later rounds) for \mathbf{s} . When the system is reordered, the reordering information is saved to unit 58 (binary) for use in continuation of a run or for fresh starts.
4. When the MME are solved, i.e., from a system already reordered, then except for the constant including $\log(2\pi)$, $-2\Lambda = \log|\mathbf{R}| + \log|\mathbf{G}| + \log|\mathbf{C}| + \mathbf{y}'\mathbf{P}\mathbf{y}$ is calculated from quantities obtained during set-up and solution of the MME:

$$\log|\mathbf{R}| = \sum_{i=1}^w N_i \log|\mathbf{R}_i|$$
 where N_i represents the number of animals having i^{th} combination of traits with $w=2^r-1$ possible combinations of r traits. The values of the w combinations of $\log|\mathbf{R}_i|$ are calculated at the beginning of MTDFLIK from the eigenvalues of $|\mathbf{R}_i|$ used to check for permissibility (see 1. above).

$$\log|\mathbf{G}| = \text{NA} \log|\mathbf{G}_0| + \text{NT} \log|\mathbf{A}|$$
 where $\log|\mathbf{G}_0|$ is calculated from the sum of logs of the eigenvalues of \mathbf{G}_0 used to check permissibility and NA is the number of animals in \mathbf{A} and NT is the total number of direct and maternal traits. As explained in Chapter 4, $\log|\mathbf{A}|$ is obtained during formation of \mathbf{A}^{-1} in MTDFNRM and read in each round of iteration from unit 44 (binary). The

term $NT\log|A|$ is a constant for a particular model of analysis, but may be needed to compare alternative models via a likelihood ratio test. If uncorrelated random effects are present, $\log|G|$ also includes $N_c \log|C_0| + \log|I|$ where N_c is the number of levels and $\log|C_0|$ is obtained from the sum of logs of eigenvalues of C_0 calculated in MTDFLIK. Note that $\log|I|$ equals zero.

$\log|C|$ is obtained in SPARSPAK during the Choleski factorization of C as explained

in Chapter 4. $y'Py$ is calculated as $y'R^{-1}y - s'W'R^{-1}y$. $y'R^{-1}y$ is accumulated by animal along with the right hand side vector $W'R^{-1}y$. SPARSPAK stores the solutions, s , in order at the beginning of the general purpose vector used by SPARSPAK so $s'W'R^{-1}y$ can be easily obtained. The SPARSNG modification described in Chapter 6 returns the solutions to the vector, SOLUT.

The value of $-2\Lambda = FVALUE$ is then returned to MTDFRUN and used in the Simplex to determine the next values of the covariance components for which to evaluate -2Λ .

Calculations of contrasts, variances of contrasts, and expected values of solutions are described in Chapter 4 and make use of the reordered and factored MME coefficient matrix.

CHAPTER SIX: Modifications to SPARSPAK to Find Constraints

In calculating the likelihood, estimating fixed and random effects, and estimating standard errors we are often faced with solving a linear system of equations. More specifically we need to solve the following set of equations

$$\mathbf{Cs} = \mathbf{r} \quad (1)$$

where \mathbf{C} is a $n \times n$ symmetric positive semi-definite matrix, \mathbf{s} and \mathbf{r} are $n \times 1$ vectors, and \mathbf{s} is unknown. Because \mathbf{C} may not be positive definite there may be more than one solution. For the cases we are interested in at least one solution is guaranteed.

The general approach that SPARSPAK takes in solving (1) is to perform a Cholesky factorization of \mathbf{C} into \mathbf{LL}' , and then to solve for \mathbf{s} using the factorization. The algorithm can be presented as follows

- (1) Factor \mathbf{C} into \mathbf{LL}' where \mathbf{L} is a lower triangular matrix.
- (2) Solve $\mathbf{Ld} = \mathbf{r}$ for \mathbf{d} .
- (3) Solve $\mathbf{L}'\mathbf{s} = \mathbf{d}$ for \mathbf{s} .

The details both for the factorization and for solving triangular systems of equations can be found for example in Stewart [1973, chap. 3].

SPARSPAK can handle the case where \mathbf{C} is positive definite but not the case where \mathbf{C} is only positive semi-definite. Alternatives when working with positive semi-definite matrices are 1) to reformulate the problem to remove the linear dependencies, 2) to add additional constraints, or 3) to modify the algorithms to detect linear dependencies. Modification of the algorithms will now be described.

Cholesky Factorization

c_{ij} = the element in row i and column j of matrix \mathbf{C} ,

r_i = the element in row i of vector \mathbf{r} ,

λ_{ij} = the element in row i and column j of matrix \mathbf{L} ,

d_i = the element in row i of vector \mathbf{d} ,

\mathbf{C}_k = the leading principal sub-matrix of order k ,

$\mathbf{C}_1 = c_{11}$

$$\mathbf{C}_k = \begin{bmatrix} \mathbf{C}_{k-1} & \mathbf{c}_k \\ \mathbf{c}'_k & c_{kk} \end{bmatrix},$$

$$\mathbf{L}_1 = \lambda_{11}$$

$$\mathbf{L}_k = \begin{bmatrix} \mathbf{L}_{k-1} & \mathbf{0} \\ \mathbf{I}'_k & \lambda_{kk} \end{bmatrix}, \text{ and}$$

$$\mathbf{C}_k = \mathbf{L}_k \mathbf{L}'_k.$$

The Cholesky factorization proceeds by sequentially finding the Cholesky factorizations (\mathbf{L}_k) of each of the leading principal sub-matrices (\mathbf{C}_k). The Cholesky factorization of the first principal sub-matrix is simply $\mathbf{L}_1 = \sqrt{\mathbf{C}_1} = \sqrt{c_{11}}$. Subsequent factorizations are found as follows:

- (1) Solve $\mathbf{L}_{k-1} \mathbf{l}_k = \mathbf{c}_k$
- (2) Set $\lambda_{kk} = \sqrt{c_{kk} - \mathbf{I}'_k \mathbf{l}_k}$.

The Cholesky factorization routine (GSFCT) within SPARSPAK -A assumes that in step 2, c_{kk} is always greater than zero which will be true if \mathbf{C} is positive definite. When \mathbf{C} is only positive semi-definite then at least one of the c_{kk} will be zero. However due to round-off the calculated $c_{kk} - \mathbf{I}'_k \mathbf{l}_k$ may not be exactly equal to zero. The first modification is to set λ_{kk} equal to zero whenever $c_{kk} - \mathbf{I}'_k \mathbf{l}_k$ is less than a tolerance parameter (τ) times the original diagonal element.

Solving Triangular Systems of Equations

The remaining modifications to SPARSPAK involve solving systems of upper and lower triangular systems of equations. Because of the great similarity in solving upper and lower triangular systems of equations, only the details for the lower triangular system are presented:

$$\mathbf{Ld} = \mathbf{r}.$$

A solution for \mathbf{d} proceeds by sequentially finding solutions to

$$\mathbf{L}_k \begin{bmatrix} d_1 \\ \vdots \\ d_k \end{bmatrix} = \begin{bmatrix} r_1 \\ \vdots \\ r_k \end{bmatrix}. \quad (2)$$

A solution for d_1 is simply a solution to $\lambda_{11} d_1 = r_1$. If $\lambda_{11} \neq 0$ then $d_1 = r_1 / \lambda_{11}$. If $\lambda_{11} = 0$ then $d_1 = 0$ will be a solution. Subsequent d_k are found as follows:

- (1) If $\lambda_{kk} = 0$ then set $d_k = 0$ as a solution.

$$(2) \text{ Otherwise } \mathbf{d}_k = \left(\mathbf{r}_k - \sum_{i=1}^{k-1} \lambda_{k-1,i} \mathbf{d}_i \right) / \lambda_{kk}.$$

The SPARSPAK routine (GSSLV) that solves a triangular system of equations assumes that λ_{kk} is always greater than zero. The only time λ_{kk} is not greater than zero is when it is set to zero in the Cholesky factorization. The remaining modifications involve simply checking when λ_{kk} is equal to zero and then setting \mathbf{d}_k equal to zero. This modification needs to be made in step 1 of the Cholesky factorization and when solving for \mathbf{s} and \mathbf{d} .

Uniqueness of the Constraints

While for positive semi-definite matrices there exist an infinite number of possible constraints, the constraints that will be found by this algorithm depend only on the order of the rows and columns in the matrix. Because the order of the rows and columns is determined once and remains fixed the constraints will remain the same unless additional linear dependencies are added or removed. For a given set of linear dependencies, the algorithm always yields the same set of constraints.

Example

The modifications will be illustrated for the following example.

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 25 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 14 \\ 53 \end{bmatrix}.$$

Starting with the Cholesky factorization of \mathbf{C} .

$$(1) \lambda_{11} = \sqrt{c_{11}} = \sqrt{1} = 1$$

$$(2) \mathbf{L}_1 \mathbf{l}_2 = \mathbf{c}_2$$

$$1 \times \mathbf{l}_2 = 2$$

$$\mathbf{l}_2 = 2/1 = 2$$

$$(3) \lambda_{22} = \sqrt{c_{22} - \mathbf{l}_2' \mathbf{l}_2} = \sqrt{4 - 2 \times 2} = 0$$

In practice if $c_{22} - \mathbf{l}_2' \mathbf{l}_2 < \tau \times c_{22}$ then set $\lambda_{22} = 0$.

$$\mathbf{L}_2 = \begin{bmatrix} 1 & 0 \\ 2 & 0 \end{bmatrix}$$

$$(4) \mathbf{L}_2 \mathbf{l}_3 = \mathbf{c}_3$$

$$\begin{bmatrix} 1 & 0 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} \lambda_{31} \\ \lambda_{32} \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \end{bmatrix}$$

$$1 \times \lambda_{31} = 3$$

$$\lambda_{31} = 3$$

$$\lambda_{32} = 0 \text{ because } \lambda_{22} = 0$$

$$\mathbf{l}_3 = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

$$(5) \lambda_{33} = \sqrt{c_{33} - \mathbf{l}_3' \mathbf{l}_3} = \sqrt{25 - 9} = 4$$

$$\mathbf{L}_3 = \mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 4 \end{bmatrix}$$

Because, λ_{22} is set equal to zero, the constraint $s_2 = 0$ has been selected for the final solution. With the Cholesky factorization of \mathbf{C} solve for \mathbf{d} :

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 4 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 14 \\ 53 \end{bmatrix}$$

$$(1) d_1 = 7/1 = 7$$

$$(2) d_2 = 0 \text{ because } \lambda_{22} = 0$$

$$(3) d_3 = (53 - 3 \times 7)/4 = 8$$

Finally, solve for \mathbf{s} :

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 0 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 0 \\ 8 \end{bmatrix}$$

$$(1) s_3 = 8/4 = 2$$

$$(2) s_2 = 0 \text{ because } \lambda_{22} = 0$$

$$(3) s_1 = (7 - 3 \times 2)/1 = 1.$$

Therefore, a solution for \mathbf{s} is

$$\mathbf{s} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}$$

Modified SPARSPAK Routines

The three SPARSPAK-A routines modified to allow for positive semi-definite matrices are SOLVE5, GSFCT, and GSSLV. The modified routines were renamed SOLV5G, GSFCTG, and GSSLVG.

CHAPTER SEVEN: A Description of the Simplex Method for DFREML

Simplex (Polytope) Method

The Simplex (polytope) method described by Nelder and Mead (1965) is a procedure designed to locate the minimum of a function with respect to several variables. The method is used in MTDFREML to locate the minimum of $-2 \log$ of the likelihood function, -2Λ , in models with multiple parameters, i.e., in models with several random effects. The (co)variance components that minimize -2Λ also maximize the log likelihood function and are REML estimates. The Simplex method is sometimes called the polytope ('many places') method to distinguish it from the simplex method for linear programming.

The polytope algorithm begins with a set of $p+1$ parameter vectors $t_0, t_1, \dots, t_{p-1}, t_p$ and their corresponding function values for -2Λ , designated $F_0, F_1, \dots, F_{p-1}, F_p$ and ordered such that $F_0 \leq F_1 \leq \dots \leq F_{p-1} \leq F_p$, where p is the number of parameters to be estimated. The parameter vectors form the vertices of a polytope or geometrical figure in p dimensions. At each round of iteration, a new polytope is formed by generating a new point to replace the worst point t_p , i.e., the point with the highest function value for $-2\Lambda = FVALUE$. The new point is generated as a linear combination of existing points by three operations: *reflection*, *expansion*, and *contraction*.

A round of iteration begins with the calculation of t_m , the centroid (center) of the best p vertices t_0, t_1, \dots, t_{p-1} from:

$$t_m = (\sum t_i) / p \quad [1]$$

for $i=0, 1, \dots, p-2, p-1$. This vector t_m consists of the average value (excluding the worst point) for each parameter to be estimated. A new point t_r is then generated by a reflection step in which the worst point t_p , i.e., the point corresponding to the largest function value, is reflected towards the center:

$$t_r = t_m + \alpha(t_m - t_p) \quad [2]$$

where α ($\alpha > 0$) is the reflection coefficient. The function -2Λ is then evaluated for t_r to yield F_r .

Three outcomes are possible:

1. $F_0 \leq F_r \leq F_{p-1}$

The reflected point t_r is neither a new best nor worst, i.e., it is intermediate. In this case, t_p is removed from the Simplex and t_r is added and the vectors are reordered such that $F_0 < F_1 \dots$

2. $F_r < F_0$

The reflected point has the smallest function value and is therefore the new best point. The direction of reflection is assumed to be good and an attempt is made to find an even better point by expanding the reflection point further in the same direction to give an expanded point t_e :

$$t_e = t_m + \gamma(t_r - t_m) \quad [3]$$

where γ ($\gamma > 1$) is the expansion coefficient. The function F_e is then evaluated with two possible outcomes:

- a) $F_e < F_r$ - the expansion was successful and t_e replaces t_p to end the round, or
- b) $F_e > F_r$ - the expansion failed so t_e is discarded and the reflected point t_r replaces t_p to end the round.

3. $F_r > F_{p-1}$

In this case the reflected point is worse than any of the beginning p points. This indicates that the reflected polytope is too large and should be contracted. One of two alternative contraction steps is used; the choice is determined by the relative values of F_r and F_p :

- a) $F_r < F_p$ - the reflected point t_r is better than the old maximum point t_p and a contracted point is calculated from:

$$t_c = t_m + \beta(t_r - t_m) \quad [4]$$

where β ($0 < \beta < 1$) is the contraction coefficient.

- b) $F_r \geq F_p$ - the reflected point t_r is not better than the old maximum point t_p and a contracted point t_c is calculated from:

$$t_c = t_m + \beta(t_p - t_m) \quad [5]$$

F_c is then calculated from [4] or [5] with two possible outcomes:

- a) $F_c < \min\{F_r, F_p\}$ - the contraction was successful and t_c replaces t_p to end the iterate, or
- b) $F_c > \min\{F_r, F_p\}$ - the contraction failed and the complete polytope is shrunk by moving each of the parameter vectors $t_1, t_2, \dots, t_{p-1}, t_p$ halfway toward the best point t_0 :

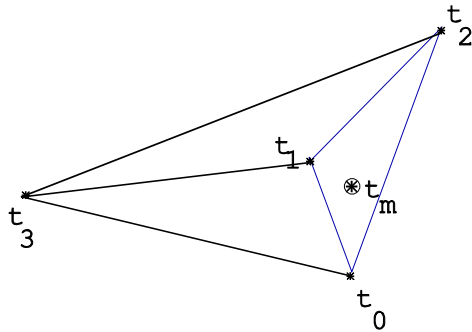
$$t'_i = (t_i + t_0) / 2 \quad [6]$$

for $i = 1, \dots, p$. The function is then evaluated for points t_1, \dots, t_p and a new round begins.

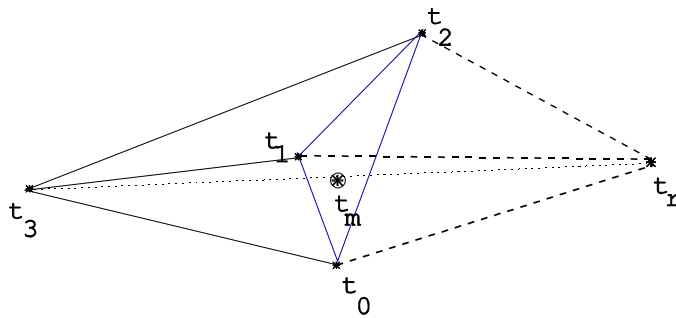
The polytope changes shape each round and moves across the likelihood function until convergence is reached. Nelder and Mead (1965) calculated the variance of the function values of the $p+1$ polytope points and stopped iteration when this variance dropped below a particular

convergence value, e.g., 1.e-6.

The basic moves of the polytope algorithm will be illustrated for a polytope in three dimensions as presented in Press et al. (1989). The polytope consists of four points and forms a tetrahedron. Let F_0 to F_3 be the function values corresponding to parameter vectors t_0 to t_3 , respectively, each consisting of values for the three parameters. The parameter vectors are ordered such that t_0 is the best (i.e., smallest FVALUE) and t_3 is the worst (i.e., largest FVALUE) point. The polytope at the beginning of the step is drawn with solid lines and the centroid of t_0 , t_1 , and t_2 is denoted t_m :

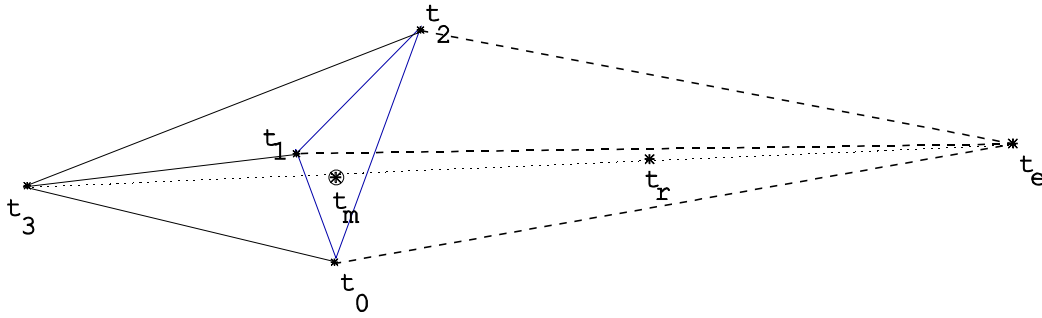


The first step is a reflection away from the worst point t_3 through t_m to t_r :



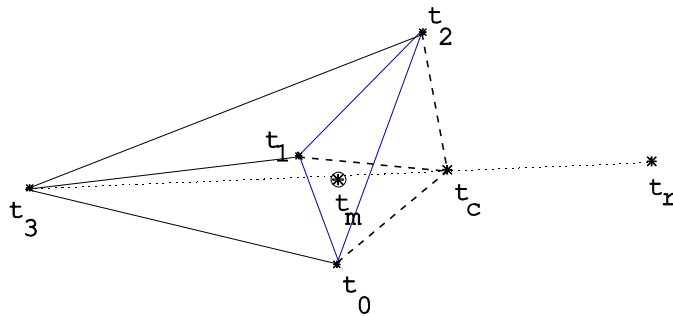
If $F_1 < F_r < F_3$, the reflected point is intermediate and replaces F_3 to form a new polytope (drawn with dashed lines) for the next round. If α , the reflection coefficient, is equal to 1, then the volume of the new polytope should be the same as the volume of the original polytope.

In the second possible outcome, the reflected point is a new minimum ($F_r < F_0$) and is further expanded away from the largest point t_3 to yield t_e :

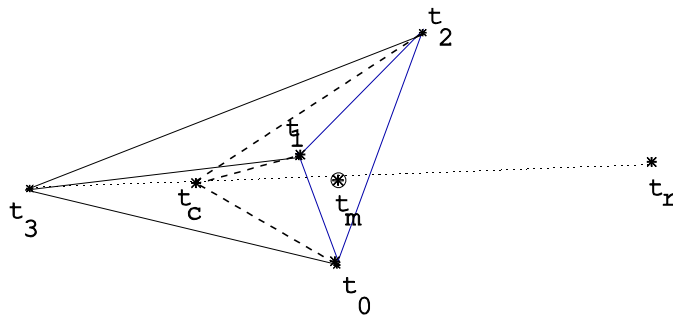


If $F_e < F_r$, the expansion was successful and t_e replaces t_3 for the next round. If $F_e > F_r$, the expansion failed and the reflected point t_r replaces t_3 . A failed expansion can result if the reflection moved the polytope into a valley but at an angle to the valley so that further expansion results in a movement up the opposite slope.

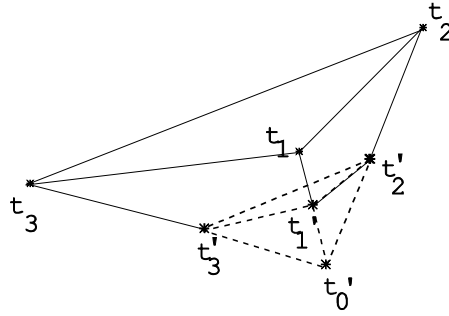
In the third possible outcome, $F_r > F_2$, i.e., the reflected point is worse than any of the remaining points. This indicates that the original polytope was at a valley floor and the reflection moved up a hill. The polytope is then contracted to t_c by moving t_r toward t_m if $F_r < F_3$:



or by moving t_3 toward t_m if $F_r \geq F_3$:



If $F_c < \min\{F_r, F_3\}$, the contraction was successful and t_c replaces t_3 . On the other hand, if $F_c > \min\{F_r, F_3\}$, the contraction failed. This is a rare event but can occur when a valley is curved and the polytope is 'unbalanced', i.e., one point is much further from the valley bottom than the others (Nelder and Mead, 1965). Contraction of the reflected point may result in a movement away from the valley bottom instead of towards it. In this situation all points of the polytope are contracted towards the smallest point and eventually all points should be brought into the valley:



The changes in the polytope volume resulting from the operations of reflection, expansion, and contraction are determined by the coefficients α , γ , and β , respectively. Nelder and Mead (1965) recommended the use of $\alpha=1.0$, $\gamma=2.0$, and $\beta=0.5$. These values are used in the subroutines presented by O'Neill (1971) and Press et al. (1989) and also in Meyer (1988).

The polytope method is applicable to the minimization of a function with respect to any number of variables, including one, but the number of iterations required for convergence is expected to increase with the number of variables. The polytope procedure will be demonstrated for an example data set consisting of measures on 282 animals (Meyer, 1989). The model used to analyze the data is the same model used to generate the data which consisted of fixed mean and generation effects and random animal (a), maternal (m), litter or common environment (c), and residual (e) effects. The assumed covariance structure for the model corresponds to Meyer's model 8:

$$\text{Var} \begin{bmatrix} \mathbf{a} \\ \mathbf{m} \\ \mathbf{c} \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} \mathbf{A}\sigma_a^2 & \mathbf{A}\sigma_{am} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}\sigma_{am} & \mathbf{A}\sigma_m^2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{NC}\sigma_c^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_n\sigma_e^2 \end{bmatrix}.$$

For this model, the vector of random effects corresponding to the general model is $\mathbf{u}'=(\mathbf{a}' \mathbf{m}' \mathbf{c}')$ and $\text{Var}(\mathbf{u})=\mathbf{G}$. The log determinant of \mathbf{G} required for evaluation of -2Λ is (e.g., Meyer,

1989):

$$\log |\mathbf{G}| = \text{NA} [\log \sigma_a^2 + \log \sigma_m^2 + \log (1-r_{am}^2)] + 2 \log |\mathbf{A}| + \text{NC} \log \sigma_c^2 \quad [7]$$

where NA is the number of animals in \mathbf{A} , r_{am} is the genetic correlation between a and m, and NC is the number of common environments (litters). SPARSPAK is used to solve the MME and provide the terms needed to calculate the portion of -2Λ that is not constant.

Note that for this single trait model, $\mathbf{R} = \mathbf{I}_N \sigma_e^2$ so the MME can be written as:

$$\begin{bmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{Z} \\ \mathbf{Z}'\mathbf{X} & \mathbf{Z}'\mathbf{Z} + \mathbf{G}^{-1} / \sigma_e^2 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{y} \\ \mathbf{Z}'\mathbf{y} \end{bmatrix}$$

$$\mathbf{C}^* \hat{\mathbf{s}} = \mathbf{r}^*$$

For this model and except for a constant (e.g., Meyer, 1989):

$$-2\Lambda = (\text{N} - \text{NF} - 2\text{NA} - \text{NC}) \log \sigma_e^2 + \log |\mathbf{G}| + \log |\mathbf{C}^*| + \mathbf{y}'\mathbf{y} - \hat{\mathbf{s}}'\mathbf{r}^*$$

where NF is the rank of $\mathbf{X}'\mathbf{X}$, and N is the number of records.

From these equations, -2Λ can be evaluated for any vector of (co)variances and the vector which minimizes the function can be determined via the polytope algorithm.

The polytope method is started with $p+1$ points defining an initial polytope where p is the number of parameters to be estimated. Because in Meyer's single trait DFREML program the residual variance is estimated by $\hat{\sigma}_e^2 = \mathbf{y}'\mathbf{y} - \hat{\mathbf{s}}'\mathbf{r}^*(\text{N} - \text{NF})$, only four components need to be estimated for the example and so the polytope consists of five points. The elements of each prior vector are expressed in DFREML as a proportion of σ_p^2 , the phenotypic variance, i.e., $\Theta_a = \sigma_a^2 / \sigma_p^2$, $\Theta_m = \sigma_m^2 / \sigma_p^2$, $\Theta_{am} = \sigma_{am} / \sigma_p^2$, and $\Theta_c = \sigma_c^2 / \sigma_p^2$, where $\sigma_p^2 = \sigma_a^2 + \sigma_m^2 + \sigma_{am} + \sigma_c^2 + \sigma_e^2$. $\lambda_j = \sigma_e^2 / \sigma_j^2$ (where the j subscript is a, m, am, or c) is required for the formation of the MME and can be found as $\lambda_j = \Theta_e / \Theta_j$ where $\Theta_e = \sigma_e^2 / \sigma_p^2 = (1 - \Theta_a - \Theta_m - \Theta_{am} - \Theta_c)$. After $\hat{\sigma}_e^2$ has been estimated, $\hat{\sigma}_p^2$ can be estimated as $\hat{\sigma}_p^2 = \hat{\sigma}_e^2 / \Theta_e$. The other (co)variance components can then be similarly obtained by multiplying the corresponding Θ_j parameter by $\hat{\sigma}_p^2$.

The user supplies a vector of p (co)variance priors for a single point and the other p points (covariance vectors) of the polytope are then generated by multiplying each of the n elements of the initial vector in turn by a step value greater than 1. A stepsize of .2 is used (new parameter = original parameter + .2 original parameter) to generate the initial polytope. Therefore, for an initial parameter vector of $\Theta_a = .40$, $\Theta_m = .15$, $\Theta_{am} = -.05$, and $\Theta_c = .10$, the initial polytope would consist of the following five points and function values:

No.	Parameter vector				-2Λ	t_i
	Θ_a	Θ_m	Θ_{am}	Θ_c		
1	.40	.15	-.05	.10	1901.24972055	1
2	.48	.15	-.05	.10	1902.54968558	4
3	.40	.18	-.05	.10	1902.10903471	3
4	.40	.15	-.06	.10	1901.00920006	0
5	.40	.15	-.05	.12	1901.82477533	2

which would be used in the first round of iteration. In order to calculate the centroid for the four best points, the initial points are ordered t_0 through t_4 such that $F_0 < F_1 < F_2 < F_3 < F_4$, i.e., from lowest (best) to highest (worst). From [1], the centroid of $t_0, t_1, t_2,$ and t_3 is $t_m = (.4 \ .1575 \ -.0525 \ .105)$. The first step is to generate a new point t_r by reflecting the worst point $t_4 = (.48 \ .15 \ -.05 \ .10)$ toward the centroid t_m according to [2]. For a reflection coefficient of $\alpha = 1.0$, the reflected point is $t_r = (.32 \ .165 \ -.055 \ .11)$ and, on calculation, $F_r = 1900.84252207$. Because the reflected point is a new best point ($F_r < F_0$), the reflected point is expanded in the same direction according to [3]. For an expansion coefficient of $\gamma = 2.0$, the expanded point is $t_e = (.24 \ .1725 \ -.0575 \ .115)$ with $F_e = 1900.65155928$. The expansion was successful, i.e., $F_e < F_r < F_0$, so t_4 is discarded, t_r is assigned to t_0 , and the former t_0 through t_3 are reassigned to t_1 through t_4 , respectively, and the first round of iteration is complete. The first round required two new likelihood evaluations, one for the reflected point and one for the expanded point, for a total of seven. The polytope at the end of round 1, ordered by function value, is:

No.	Parameter vector				-2Λ	t_i
	Θ_a	Θ_m	Θ_{am}	Θ_c		
7	.24	.1725	-.0575	.115	1900.65155928	0
4	.40	.15	-.06	.10	1901.00920006	1
1	.40	.15	-.05	.10	1901.24972055	2
5	.40	.15	-.05	.12	1901.82477533	3
3	.40	.18	-.05	.10	1902.10903471	4

The variance of the five function values corresponding to the new points of the polytope is .35344680. The second round begins with the calculation of a new centroid $t_m = (.36 \ .155625 \ -.054375 \ .10875)$ and reflected point $t_r = (.32 \ .13125 \ -.05875 \ .1175)$. For the example data, the reflection point of the

second round is a new best point ($F_r=1900.36183550$) and the subsequent expansion with $t_e=(.28$
 $.106875 \ -0.063125 \ .12625)$ is successful so $t_e=t_0$ is added to the polytope instead of t_r and t_4 is
discarded. The polytope at the beginning of round three is:

No.	Parameter vector				-2Λ	t_i
	Θ_a	Θ_m	Θ_{am}	Θ_c		
9	.28	.106875	-.063125	.12625	1900.29696043	0
7	.24	.1725	-.0575	.115	1900.65155928	1
4	.40	.15	-.06	.10	1901.00920006	2
1	.40	.15	-.05	.10	1901.24972055	3
5	.40	.15	-.05	.12	1901.82477533	4

and the centroid is $t_m=(.33 \ .14484375 \ -.05765625 \ .1103125)$. The reflected point is $t_r=(.26$
 $.1396875 \ -.0653125 \ .100625)$ with $F_r=1900.26603219$, a new best point. The expanded point is
 $t_e=(.19 \ .13453125 \ -.07296875 \ .0909375)$ with $F_e=1901.65701486$. Because $F_e > F_r$, the expansion
is unsuccessful and $t_e=t_0$ is added to the polytope and t_4 is deleted to complete round three. Rounds
4 to 6 of iteration proceed as follows:

a) Round 4 -

Initial polytope:

No.	Parameter vector				-2Λ	t_i
	Θ_a	Θ_m	Θ_{am}	Θ_c		
10	.26	.1396875	-.0653125	.100625	1900.26603219	0
9	.28	.106875	-.063125	.12625	1900.29696043	1
7	.24	.1725	-.0575	.115	1900.65155928	2
4	.40	.15	-.06	.10	1901.00920006	3
1	.40	.15	-.05	.10	1901.24972055	4

centroid $t_m=(.25000000 \ .14226563 \ -.06148438 \ .11046875)$

reflection $t_r=(.19 \ .13453125 \ -.07296875 \ .12093750)$

with $F_r=1901.17966292$

Because $F_r > F_3$, a contraction is required. The reflected point is better than the worst point, i.e., F_r
 $< F_4$ so equation [4] is used with a contraction coefficient of $\beta=0.5$ to calculate the contraction point

$t_c=(.2425 \ .13839844 \ -.06722656 \ .11570313)$ with $F_c=1900.40869337$. Because $F_c < F_r=\min\{F_r, F_4\}$, the contraction is successful, $t_c=t_2$ is added to the polytope and t_4 is deleted.

b) Round 5 -

Initial polytope:

No.	Parameter vector				-2Λ	t_i
	Θ_a	Θ_m	Θ_{am}	Θ_c		
10	.26	.13968750	-.0653125	.100625	1900.26603219	0
9	.28	.106875	-.063125	.12625	1900.29696043	1
13	.2425	.13839844	-.06722656	.11570313	1900.40869337	2
7	.24	.1725	-.0575	.115	1900.65155928	3
4	.40	.15	-.06 .10	1901.00920006	4	

centroid $t_m=(.2556250 \ .13936523 \ -.06329102 \ .114394531)$

reflection $t_r=(.111250 \ .12873047 \ -.06658203 \ .12878906)$

with $F_r=1903.41542350$

Because $F_r > F_3$, a contraction is required. The reflected point is worse than the worst point, i.e., $F_r > F_4$, so equation [5] is used to calculate the contraction point $t_c=(.3278125 \ .14468262 \ -.06164551 \ .10719727)$ with $F_c=1900.38019626$. Because $F_c < F_4=\min\{F_r, F_4\}$, the contraction is successful, $t_c=t_2$ is added to the polytope and t_4 is deleted.

c) Round 6 -

Initial polytope:

No.	Parameter vector				-2Λ	t_i
	Θ_a	Θ_m	Θ_{am}	Θ_c		
10	.26	.13968750	-.0653125	.100625	1900.26603219	0
9	.28	.106875	-.063125	.12625	1900.29696043	1
15	.3278125	.14468262	-.06164551	.10719727	1900.38019626	2
13	.2425	.13839844	-.06722656	.11570313	1900.40869337	3
7	.24	.1725	-.0575	.115	1900.65155928	4

centroid $t_m=(.27757812 \ .13241089 \ -.06432739 \ .12443848)$

reflection $t_r=(.31515625 \ .09232178 \ -.07115479 \ .10988770)$

with $F_r=1900.39163330$

The reflected point t_r is intermediate, i.e., $F_2 < F_r < F_3$, so it is retained and t_4 is deleted. At the end of six rounds, 16 likelihoods have been evaluated and the polytope is:

No.	Parameter vector				-2Λ	t_i
	Θ_a	Θ_m	Θ_{am}	Θ_c		
10	.26	.13968750	-.0653125	.100625	1900.26603219	0
9	.28	.106875	-.063125	.12625	1900.29696043	1
15	.3278125	.14468262	-.06164551	.10719727	1900.38019626	2
16	.31515625	.09232178	-.07115479	.10988770	1900.39163330	3
13	.2425	.13839844	-.06722656	.11570313	1900.40869337	4

and the variance of the function values is .00398640. If this value is less than the convergence criterion, then t_0 , the parameter vector with the minimum function value, would be used to estimate the variance components. For this parameter vector, residual variance estimated from $\hat{\sigma}_e^2 = \mathbf{y}'\mathbf{P}\mathbf{y}/(N - NF)$ is $\hat{\sigma}_e^2 = 53.53557453$ and $\Theta_e = (1 - \Theta_a - \Theta_m - \Theta_{am} - \Theta_c) = .565$. Therefore, $\hat{\sigma}_p^2 = \hat{\sigma}_e^2 / \Theta_e = 94.75322925$, $\hat{\sigma}_a^2 = \hat{\sigma}_p^2 \Theta_a = 24.63583960$, $\hat{\sigma}_m^2 = \hat{\sigma}_p^2 \Theta_m = 13.23584171$, $\hat{\sigma}_{am}^2 = \hat{\sigma}_p^2 \Theta_{am} = -6.188570285$, and $\hat{\sigma}_c^2 = \hat{\sigma}_p^2 \Theta_c = 9.534543693$. If a smaller convergence criterion was desired, then reflection, expansion, and contraction steps would continue. The following results were obtained for smaller convergence criteria:

Convg. Crit.	Number		Parameters				
	Rounds	Likeli.	Θ_a	Θ_m	Θ_{am}	Θ_c	$\hat{\sigma}_e^2$
10^{-5}	17	37	.308667	.132939	-.065397	.094833	50.839485
10^{-6}	22	47	.315677	.144102	-.066962	.082729	50.513202
10^{-7}	27	56	.318319	.139051	-.066477	.086184	50.373084
10^{-8}	30	62	.314638	.140445	-.066394	.086756	50.529405
10^{-9}	36	73	.316798	.139305	-.066488	.086865	50.432655

These results agree with Meyer (1989) who reported that a convergence criterion of 1.e-5 to 1.e-6 is usually sufficient in terms of changes in parameter estimates.

The polytope search procedure can now be evaluated in terms of the three desirable characteristics of a search strategy listed by Meyer (1989): a) number of likelihood evaluations required, b) robust and relatively free from problems of numerical accuracy, and c) accomodate constraints on the parameter space.

Likelihood Evaluations Required

After formation of the initial polytope which requires $p+1$ likelihood evaluations, each round of iteration requires one or two likelihood evaluations unless a contraction is unsuccessful and the polytope must be shrunk; in this situation, $p+2$ likelihoods must be evaluated. Fortunately, a failed contraction is very rare (Nelder and Mead, 1965). For the example illustrated, a convergence criterion of 1.e-9 required 36 rounds of iteration and 73 likelihood evaluations as follows:

Polytope operation	No.	×	Likelihoods/operation	=	Total likelihoods
initial polytope	1		5		5
reflection only	4		1		4
reflection, expansion	6		2		12
reflection, contraction	26		2		52
reflection, contraction, shrinkage	0		6		0
					<hr/> 73 <hr/>

Therefore, most rounds required two evaluations and a failed contraction did not occur. Meyer (1989) found that starting values close to the final estimates reduced the number of rounds required to reach convergence and suggested that a small stepsize might be more efficient for good starting values. Compared to a stepsize of 0.2, a stepsize of 0.1 for the same initial parameter values in the example data required three fewer polytope iterations and eight fewer likelihood evaluations to reach a convergence criterion of 1.e-9. O'Neill (1971) stated that the polytope algorithm will work successfully for all starting values and stepsizes but those far from optimum will require more iteration to reach convergence. Press et al. (1989) note that other multidimensional minimization algorithms, e.g., Powell's method (see Chapter 8), may be faster but the polytope method is often preferred because of its simplicity and generality.

Robust and Relatively Free from Problems of Numerical Accuracy

According to Gill et al. (1981), minimization techniques such as the polytope method which are based on function comparison are not susceptible to rounding errors but few guarantees can be made concerning convergence. A potential problem for all minimization methods, especially with multiple parameters, is false convergence to a point other than the minimum. On the basis of limited investigation, Meyer (1989) reported that while the number of likelihoods required for convergence varied with the starting values, final estimates were invariant to the starting values selected and suggested that local maxima are not a problem with REML estimation via the DF algorithm. Kovac and Groeneveld (1989) used the polytope procedure in a multiple trait animal model to estimate additive genetic and residual covariance matrices of two traits (backfat thickness and daily gain) from 649 field test records of boars. They found that different starting values converged to two distinct estimated parameter sets and concluded that local maxima can exist in multivariate data sets. Boldman and Van Vleck (1990), working with the simulated data of Meyer (1989) found that different direct-maternal correlation (r_{am}) priors converged to different final estimates. They found that the EM algorithm, however, converged to the same final estimates which indicated that the variation in estimates for the DF algorithm may result from the failure of the polytope algorithm to locate the global maximum rather than the existence of local maxima. Further study (Boldman and Van Vleck, Chapter 8) indicated that when the analyses utilizing the polytope method were restarted at the claimed minimum with an initial stepsize of 0.2 then the same final parameter vector was obtained in all analyses. The increased reliability of the final estimates would seem to be well worth the additional likelihood evaluations required for a restart. For this restart, a reduced stepsize, i.e., < 0.2 , is likely to be sufficient (MTDFREML can be easily modified to utilize the smaller stepsize).

A technique used to test for convergence to a local minimum in the polytope subroutine described by O'Neill (1971) is to calculate the function values for $2p$ points consisting of the predicted minimum t_0 plus and minus 0.001 times the original stepsize. If all $2p$ of these values are greater than F_0 , then t_0 is accepted as the minimum. If one or more of the $2p$ points has a function value less than F_0 , then the initial convergence point was a local minimum. In this case, the polytope is contracted around the new lowest point and restarted. Modification of MTDFREML to utilize this convergence criterion should be straight forward. The polytope method should be compared with other multidimensional minimization algorithms to determine if the polytope method is best in terms of number of likelihood evaluations required and reliability of estimates.

Accommodate Constraints on the Parameter Space

Animal models with several random effects require several constraints on the parameter space. For example, in the model described above, parameters Θ_a , Θ_m , Θ_c , and the absolute value of parameter Θ_{am} must be between 0 and 1 in addition to the sum of the parameters. Because the genetic correlation between direct and maternal effects must be between -1 and 1, this also implies that $\Theta_{am}^2 \leq \Theta_a \Theta_m$. Even if the vertices of the initial polytope are within the parameter space, subsequent reflection or expansion steps can move parameters outside the parameter space. In MTDFREML, before the likelihood is evaluated for a particular set of parameters a check is made to determine if all parameters are within the parameter space. If not, then the likelihood function is not evaluated but instead, a large positive function value (1.e+37) is assigned to the parameter vector. As previously illustrated, if the function value of a new point is worse (higher) than any of the remaining points then the polytope will be contracted away from the high point. Therefore, any movement of the polytope outside of the parameter space will be followed by one or more contraction steps which will eventually move estimates back within the bounds. One of the advantages of the polytope method for multidimensional minimization is the ease with which restrictions on the parameter space can be imposed.

CHAPTER EIGHT: Comparison of Simplex and Powell's Search Methods for DFREML

Introduction

REML estimation of (co)variance components is generally considered the best method for unbalanced animal breeding data. For a particular data set and assumed model, REML estimation involves maximization of the log of the likelihood function (Λ) of the data which is independent of any fixed effects and includes a nonlinear function of the (co)variance components. The values of the (co)variances within the allowable parameter space which maximize Λ , or equivalently minimize -2Λ , are the REML estimates. Several different REML algorithms have been developed but most methods are iterative and require formation and manipulation of the mixed model equations. Iterative procedures to determine the maximum of a function can be assigned to one of two general classes: gradient methods, which utilize partial derivatives of the objective function; or direct search methods, which use only evaluations of the function.

Most REML algorithms commonly used in animal breeding research are gradient methods, e.g., the method of scoring utilizes expected values of second derivatives of Λ and the expectation-maximization (EM) algorithm requires expected values of first derivatives. A major limitation of the gradient-type algorithms is that they require inversion of the coefficient matrix of the MME. Traditional dense matrix inversion algorithms are of limited use with an animal model (AM) where the order of the MME often exceeds the number of records. Misztal (1990) demonstrated, however, that storage and time requirements for an EM-type REML algorithm can be greatly reduced by the use of sparse matrix methods in which operations on zero elements are avoided.

A direct search type algorithm to obtain REML estimates of variance components in an AM with only one random effect was presented by Graser et al. (1987). In this method, termed the derivative-free (DF) approach, Λ and the residual component of variance are obtained via Gaussian elimination of the mixed model coefficient matrix augmented by the right hand side and the sum of squares of the data. The method requires only a one-dimensional search because Λ is maximized with respect to a single parameter, the ratio of animal variance to residual variance, $r = \sigma_a^2 / \sigma_e^2$. For the one-dimensional search, Graser et al. (1987) repeatedly fit a quadratic in r to Λ and used the maximum of the estimated quadratic to obtain a new value of r to maximize Λ . They reported that convergence usually required only a few likelihood evaluations. The main advantage of the DF method over the EM algorithm is that the former requires solution instead of inversion of the MME. Misztal (1990)

reported that with sparse-matrix techniques, Gaussian elimination of the MME for a typical AM requires only about 5% of the time needed to obtain a full inverse.

The DF algorithm is flexible and can be extended, for example, to models with several random effects in which Λ is maximized with respect to several parameters. Meyer (1988) developed a set of computer programs, named DFREML, for parameter estimation in single trait animal models with up to three random effects and five (co)variance components. The Simplex method (Nelder and Mead, 1965) was selected as the multidimensional direct search method because of its ease of use, lack of numerical accuracy problems, and ability to accommodate constraints on the parameter space.

A potential problem with all maximization methods, especially for multiple parameters, is convergence to a point other than the global maximum. On the basis of limited investigation, Meyer (1989) reported that, while the number of likelihood evaluations required for convergence varied, parameter estimates obtained via the Simplex method were invariant to the starting values selected. More recently, convergence to different estimates when using different starting values for the same data was noted for the Simplex method in a multiple trait AM (Groeneveld and Kovac, 1990) and an AM with correlated direct and maternal genetic effects (Luis Gama, personal communication, 1990). Misztal (1990) suggested that sparse matrix EM-REML might be preferable to DFREML methods because of potential problems with false maxima. Alternative multidimensional direct search algorithms may be better suited to maximization of Λ with DFREML methods. For example, Powell's method (1964) is generally considered to be more efficient than the Simplex method (Box et al., 1969; Press et al., 1989) but the latter may be more robust (Parkinson and Hutchinson, 1972). The purpose of this study was to compare, in terms of robustness and efficiency, the Simplex method (SM) and Powell's method (PM) for maximization of Λ in an AM with correlated direct and maternal genetic effects.

Multidimensional Direct Search Algorithms

Most direct search algorithms locate the minimum of a function but in REML can be maximized by locating the minimum of $-\Lambda$ or -2Λ . Two general classes of multidimensional direct search algorithms are sequential methods and linear methods (e.g., Jacoby, et al., 1972). In the first category the objective function is evaluated at the vertices of a geometric figure in the parameter space, and the figure is moved toward the function minimum. Linear methods generate direction vectors to move from the current set of estimates toward the minimum. Box et al. (1969) stated that SM is the most efficient sequential method and that PM is probably the most effective linear method. The use of these algorithms to locate the minimum of a function such as $-\Lambda$ will be outlined.

Simplex Method

A Simplex is a geometric figure formed by a set of $n+1$ points in n -dimensional space. For example, a Simplex of two dimensions is a triangle and of three dimensions is a tetrahedron. In SM, the n -dimensional space is the parameter space of the n independent variables and each of the $n+1$ vertices of the Simplex is a parameter vector of length n and its corresponding function value, e.g., $-A$. In each round of iteration, a new Simplex is formed by generating a new point to replace the worst, i.e., the point with the largest function value, so the Simplex gradually moves "downhill" through the parameter space toward the minimum.

The vertices of the initial Simplex are parameter vectors of length n designated \mathbf{t}_i for $i=0, 1, 2, \dots, n$ and their corresponding function values F_i . At the beginning of each round, the points are ordered so that $F_0 < F_1 < \dots < F_{n-1} < F_n$, i.e., the function value for parameter vector \mathbf{t}_0 is the smallest and for \mathbf{t}_n is the largest. Because the search is for the minimum of the function, \mathbf{t}_n is the worst vertex and will be replaced to form a new Simplex. The single new point is generated as a linear combination of existing points by three operations: reflection, expansion, and contraction. See Chapter 7 for a more complete description of the Simplex method.

Powell's Method

Linear methods to minimize a function of n independent variables generally utilize a series of one-dimensional minimizations in each of n directions. Most methods initially search along the coordinate directions but differ in the procedure used to generate new directions. In PM an attempt is made to generate after n rounds of iteration a set of n direction vectors which are mutually conjugate or non-interfering, so any reduction in the function value obtained during minimization in one direction is not lost when searching in a subsequent direction. A set of mutually conjugate search directions will minimize a quadratic function in a single round of linear searches along the directions and is quite efficient even for non-quadratic functions (Press et al., 1989). PM also assures that the direction vectors do not become linearly dependent because dependent directions will find the minimum of the function only over a subspace of the full n -dimensional parameter space. Thus PM to locate the minimum of a function is both relatively efficient and, more importantly, accurate.

Each round of iteration for PM begins with a parameter vector \mathbf{p}_0^k , which like \mathbf{t}_i of SM consists of the current value for each of the n independent variables, and a set of n direction vectors $\xi_1^k, \xi_2^k, \dots, \xi_n^k$ each of length n where the superscript k indicates the current round of iteration and the subscript is the search direction. The value of the function at this starting point \mathbf{p}_0^k is denoted F_0^k and the objective is to search in turn along each of the n directions to find new values of the independent variables

which will result in a smaller function value. For each direction ξ_r^k for $r=1, 2, \dots, n$, a one-dimensional search is conducted to determine the scalar step length λ_r^k so that the function evaluated at $(\mathbf{p}_{r-1}^k + \lambda_r^k \xi_r^k)$ is a minimum along direction ξ_r^k , and $\mathbf{p}_r^k = (\mathbf{p}_{r-1}^k + \lambda_r^k \xi_r^k)$ then becomes the starting point for the next direction of the current round. At the end of the round, after all n directions have been searched, the starting point \mathbf{p}_0^k has been moved to \mathbf{p}_n^k which has a function value denoted F_n^k with $F_n^k < F_0^k$, i.e., it is closer to the minimum.

In PM the initial search directions are parallel to the coordinate axes, i.e., columns of an identity matrix, and thus linearly independent but an attempt is made to generate a new conjugate search direction at the end of each round to replace one of the old directions. If a new direction is added, it is defined as $\xi = (\mathbf{p}_n^k - \mathbf{p}_0^k)$, the vector of total progress in the iteration. This new direction vector ξ could replace any of the n old directions, but Powell (1964) shows that the proper action is to discard the direction vector along which the function made its largest decrease in the previous round. As pointed out by Press et al. (1989), this direction to be replaced, denoted ξ_m , is likely to be a major component of the new direction ξ , so discarding ξ_m avoids buildup of linear dependence.

Under certain conditions, however, it is better not to add a new search direction, but instead to retain the old direction vectors for the next round. Powell's advice is to add the new direction vector ξ only if the resulting set of direction vectors remain linearly independent, i.e., span the entire n -dimensional parameter space. The criterion requires function values obtained during the current round of iteration. Along with F_0^k and F_n^k , the function values at the beginning and end of the current round, two additional quantities are required. First, Δ is defined to be the magnitude of the function decrease along ξ_m , the direction of largest decrease in the current round, i.e., $\Delta = (F_{m-1}^k - F_m^k)$. Second, define F_e^k to be the function value at an expanded point $\mathbf{p}_e^k = \{\mathbf{p}_n^k + (\mathbf{p}_n^k - \mathbf{p}_0^k)\}$, which is the endpoint \mathbf{p}_n^k moved a further step equal to $(\mathbf{p}_n^k - \mathbf{p}_0^k)$, the distance moved in the current round. Because $\xi = (\mathbf{p}_n^k - \mathbf{p}_0^k)$ is the proposed new direction, the expanded point indicates the efficiency of the new search direction. After calculation of Δ and F_e^k , a test is performed to determine if ξ_m should be replaced by the new direction vector ξ . The criterion, derived by Powell (1964), is based on the determinant of a function of the search directions which will increase if a more efficient yet independent search direction is added. Specifically, Powell shows that if either $F_e^k \geq F_0^k$ and/or $\{2(F_0^k - 2F_n^k + F_e^k)(F_0^k - F_n^k - \Delta)^2\} \geq \{\Delta(F_0^k - F_e^k)^2\}$ then the old search directions should be retained for the next round, i.e., $\xi_r^{k+1} = \xi_r^k$ for $r=1, \dots, n$ and the endpoint \mathbf{p}_n^k is used for \mathbf{p}_0^{k+1} , the starting point for the next round. Otherwise the step length λ is found so that the function evaluated at $(\mathbf{p}_n^k + \lambda \xi)$ is a minimum along the new direction ξ and this point is then used as \mathbf{p}_0^{k+1} . For the next iterate, ξ_m^k , the direction of largest decrease in the previous round, is

discarded from the list and ξ is added at the end, that is $(\xi_1^{k+1}, \xi_2^{k+1}, \dots, \xi_n^{k+1}) = (\xi_1^k, \xi_2^k, \dots, \xi_{m-1}^k, \xi_{m+1}^k, \dots, \xi_n^k, \xi)$. The search is continued in the next round for the new starting point and set of direction vectors. Convergence is assumed when the values for the independent variables between successive iterations are less than the required accuracy. Because one of the previously generated conjugate directions may be replaced in a subsequent round, or the same set of directions may be reused, a set of n mutually conjugate directions is rarely obtained after n rounds. As a result, the number of function evaluations required to reach convergence may be increased, but this is required to insure the accuracy of the estimates. A flow diagram illustrating the basic steps of PM is given in Figure 1.

The unidimensional search used in PM to locate the value of λ_r^k for $r=1, \dots, n$ which minimizes the function through the point \mathbf{p}_{r-1}^k in direction ξ_r^k is based on a quadratic approximation to the function. In this iterative method a quadratic equation $q_2\lambda^2 + q_1\lambda + q_0 = F(\lambda)$ is fit where $F(\lambda)$ are the function values corresponding to the three current values of λ . The three initial values of λ should be chosen so that the three (co)variance vectors $(\mathbf{p}_{r-1}^k + \lambda_r^k \xi_r^k)$ span the parameter space. The minimum of the estimated parabola gives a new value for λ denoted λ' which, along with its function value, replaces the λ furthest away in the previous set of three. The process is then repeated beginning with the calculation of a new parabola. Convergence is assumed when the λ' predicted from the quadratic differs by less than the required accuracy from the λ corresponding to the smallest function value in the current set of three. Powell (1964) presents several refinements of the basic quadratic approximation method which ensures stable and efficient convergence.

Like SM, PM is usually classified as an unconstrained minimization method but can accommodate constraints on the parameter space by assigning a large function value to any non-permissible parameter vector. Even if the starting point for an iterate \mathbf{p}_0^k is within the parameter space, subsequent actions can move parameters out of the parameter space. For example, during a linear minimization the predicted minimum of the parabola may correspond to a point outside the range of the three existing points. If this point is outside of the parameter space and is assigned a large positive function value, then the new step length predicted from the parabola fit through this point and the two old points adjacent will automatically be moved away from the non-permissible vector. In addition, the expanded point $\mathbf{p}_e^k = \{\mathbf{p}_n^k + (\mathbf{p}_n^k - \mathbf{p}_0^k)\}$ used to determine if a new direction should be added can move outside of the parameter. If a large function value is assigned to F_e^k in this situation, then $F_e^k \geq F_0^k$ and the old search directions will be retained for the next round.

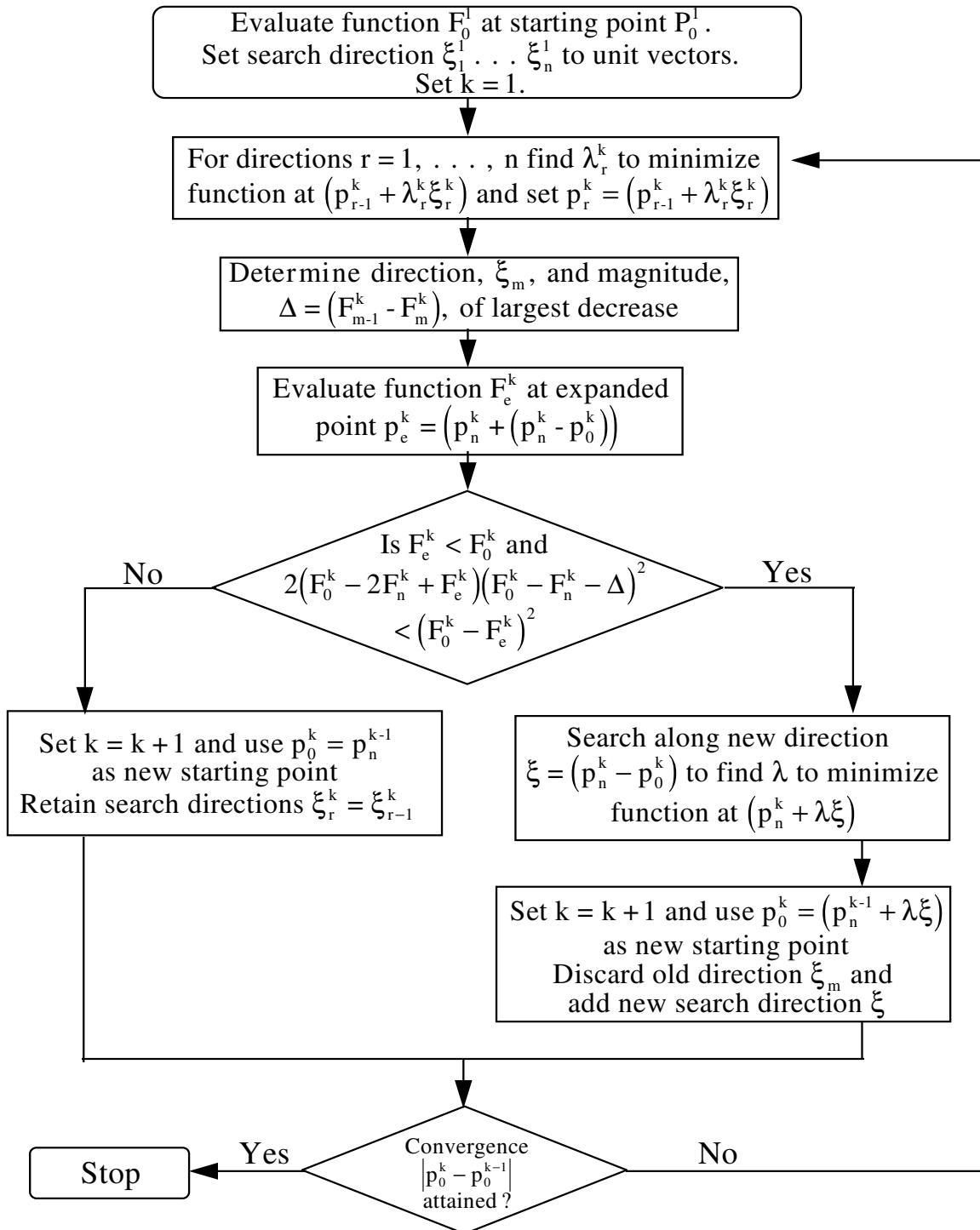


Figure 1. Flowchart for Powell's Method

A Numerical Comparison

Data

A simulated data set (see Chapter 7) from Meyer (1989) with correlated direct and maternal effects was used to compare the performance of SM and PM. Data consisted of 282 animals with records in two generations. Each generation consisted of 18 full-sib families of size 6 to 10, with each sire mated to three dams. The 24 parents of generation one (base animals) did not have records, yielding a total of 306 animals. Records were generated from an AM with a mean of 200, a fixed generation effect with two levels assigned values of 0 and 20, and random additive direct animal (a), additive maternal (m), common environment (c) (e.g., litter) and residual effects with associated (co)variances $\sigma_a^2=40$, $\sigma_m^2=15$, $\sigma_{am}=-5$, $\sigma_c^2=10$, and $\sigma_e^2=50$. The assumed covariance structure for the model was:

$$\text{Var} \begin{bmatrix} \mathbf{a} \\ \mathbf{m} \\ \mathbf{c} \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} \mathbf{A}\sigma_a^2 & \mathbf{A}\sigma_{am} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}\sigma_{am} & \mathbf{A}\sigma_m^2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{\text{NC}}\sigma_c^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_n\sigma_e^2 \end{bmatrix},$$

where \mathbf{A} is the numerator relationship matrix of order 306, and the identity matrices are of order 36 and 282 for σ_c^2 and σ_e^2 , respectively.

Analysis

The model used to estimate (co)variance components via DFREML was the same AM used to generate the data. Because residual variance can be estimated directly from terms arising during formation and solution of the MME, the direct search to maximize Λ was with respect to only four components σ_a^2 , σ_m^2 , σ_{am} , and σ_c^2 . The four elements of each prior vector are expressed in Meyer's (1988, 1989) DFREML program as a proportion of the phenotypic variance, σ_p^2 , i.e., $\Theta_a=\sigma_a^2/\sigma_p^2$, $\Theta_m=\sigma_m^2/\sigma_p^2$, $\Theta_{am}=\sigma_{am}/\sigma_p^2$, and $\Theta_c=\sigma_c^2/\sigma_p^2$, where $\sigma_p^2=\sigma_a^2+\sigma_m^2+\sigma_{am}+\sigma_c^2+\sigma_e^2$. Thus, Θ_a and Θ_m correspond to heritability of direct effects and maternal effects, h_a^2 and h_m^2 , respectively. After $\hat{\sigma}_e^2$ has been estimated, $\hat{\sigma}_p^2$ can be estimated as $\hat{\sigma}_p^2=\hat{\sigma}_e^2/\Theta_e$ where $\Theta_e=\hat{\sigma}_e^2/\hat{\sigma}_p^2=(1-\Theta_a-\Theta_m-\Theta_{am}-\Theta_c)$. The other (co)variance components can be obtained by multiplying the corresponding Θ by $\hat{\sigma}_p^2$. Starting values $\Theta_a=.40$, $\Theta_m=.15$, and $\Theta_c=.10$ were used in each analysis but the starting value of Θ_{am} was changed for each of 18 analyses, resulting in a prior for r_{am} , the genetic correlation between direct and maternal effects, which ranged from -.98 to .98.

Several researchers have presented variants of SM or PM method with modifications designed to increase efficiency and/or robustness. For this investigation, however, the original version of each method was used. The SM as described by Nelder and Mead (1965) with reflection, expansion, and

contraction coefficients $\alpha=1.0$, $\gamma=2.0$, and $\beta=.5$ is used in the MTDFREML program and in Meyer's (1988) DFNRM program. The program is supplied with a vector of n Θ parameters for a single vertex and the other n vertices of the Simplex are then generated by multiplying each of the n elements of the initial vector in turn by a step size value greater than 0. MTDFREML program uses a step size of .20 to generate the original Simplex, i.e., new parameter = original parameter + .2 (original parameter). For the model used with four independent parameters, the Simplex consists of five points. Therefore, for an initial parameter vector of $(\Theta_a \ \Theta_m \ \Theta_{am} \ \Theta_c)=(.40 \ .15 \ -.05 \ .10)$, the original Simplex would consist of this vector and four other vectors $(.48 \ .15 \ -.05 \ .10)$, $(.40 \ .18 \ -.05 \ .10)$, $(.40 \ .15 \ -.06 \ .10)$, and $(.40 \ .15 \ -.05 \ .12)$. Several rounds of iteration consisting of reflection, expansion and contraction steps were then conducted until the variance of -2Λ for the five vertices of the Simplex was less than $1.e-9$ which is smaller and thus more demanding than the value of $1.e-8$ recommended by Meyer (1988).

The implementation of PM presented by Kuester and Mize (1973) was used in the analyses after correction of an error (Karin Meyer, personal communication, 1990) in the published code. The convergence criterion was set to an accuracy $1.e-1$ (Kuester and Mize, 1973) and as recommended by Powell (1964) the procedure was continued until an iteration caused the change in each independent variable to be less than one-tenth of the required accuracy or, i.e., $1.e-2$. The initial step size for the linear search was set to .01 and the linear search was continued until λ' , the step size predicted from the quadratic, differed by less than 3% from the step size corresponding to the smallest function value in the current set of three. For example, assume the original parameter vector is $\mathbf{p}_0^1=(\Theta_a \ \Theta_m \ \Theta_{am} \ \Theta_c)=(.40 \ .15 \ -.05 \ .10)$ and the function value for this vector is F_0^1 . Because the original search directions are parallel to the coordinate axes, $\xi_1^1=(1 \ 0 \ 0 \ 0)$ and the other two points for the first quadratic would then be $(\mathbf{p}_0^1+.01\xi_1^1)=(.41 \ .15 \ -.05 \ .10)$ and either $(\mathbf{p}_0^1+.02\xi_1^1)=(.42 \ .15 \ -.05 \ .10)$ or $(\mathbf{p}_0^1-.01\xi_1^1)=(.39 \ .15 \ -.05 \ .10)$, depending on whether the function value at $(.41 \ .15 \ -.05 \ .10)$ is less than or greater than F_0^1 , respectively. These three function values and step lengths are then used in the first quadratic to predict the step length λ' which minimizes the function. The search in the first direction is terminated when $(|\lambda'-\lambda_m|/\lambda') \leq .03$ where λ_m is the step size corresponding to the smallest function value in the former set of three.

Robustness

In a comparison of alternative direct search procedures the most important criterion is the reliability of estimates obtained. According to Gill et al. (1981) direct search methods are not generally susceptible to rounding errors associated with complicated numerical processes but few if any guarantees can be made concerning convergence. Ideally, for a given data set, convergence to the same point should result from any prior parameter vector.

Simplex Method

Estimates obtained at convergence with SM were not invariant to the prior value for Θ_{am} (Table 1). For seven of the Θ_{am} priors, the minimum of $-\Lambda$ (omitting constants) was identified as 950.060 corresponding to $\Theta_a=.33$, $\Theta_m=.16$, $\Theta_{am}=-.09$, and $\Theta_c=.08$. This point was accepted as the global minimum because a larger value for $-\log \Lambda$ was obtained for all other parameter estimates. The other nine solutions indicate failure of SM to locate the global minimum for all priors. Jacoby et al. (1972) point out that false convergence can result if the Simplex contracts into a valley and stops prematurely, resulting in convergence to a local rather than a global minimum. Continuation of iteration to a smaller convergence criteria may only contract the Simplex further into the valley. Press et al. (1989) suggested that any multidimensional minimization procedure should be restarted at the claimed minimum. If, after initial convergence, the Simplex is regenerated around the best point \mathbf{t}_0 then one or more of the new vertices may be moved out of the valley and subsequent steps may move the Simplex to the global minimum. Restarts of SM using the initial converged values as priors resulted in the same final estimates for all initial priors except .10 (Table 3). The analysis for the Θ_{am} prior .20 was not restarted because the estimates at initial convergence were the same as the priors, i.e., the Simplex did not move.

A restart of SM was effective for most occurrences of false convergence but is inefficient if the values at initial convergence are at the global minimum. For example, restarting the procedure with the estimates $\Theta_a=.33$, $\Theta_m=.16$, $\Theta_{am}=-.09$, and $\Theta_c=.08$ and a step size .20 resulted in the same estimates at reconvergence but 42 additional likelihood evaluations were required. Ideally, a criterion exists which could be used to test for convergence to a local minimum. In the Simplex subroutine presented by O'Neill (1971), after convergence the function value is calculated at the $2n$ points consisting of the suspected minimum \mathbf{t}_0 plus and minus .001 times the original step size along the axis corresponding to each of the n independent variables. For example, if the suspected minimum was $(\Theta_a=.33, \Theta_m=.16, \Theta_{am}=-.09, \text{ and } \Theta_c=.08)$ and the original step size was .20 then $-\Lambda$ would be evaluated at the eight points $(.33 \pm .0002, .16, -.09, .08)$, $(.33, .16 \pm .0002, -.09, .08)$, $(.33, .16, -.09 \pm .0002, .08)$, and $(.33, .16, -.09, .08 \pm .0002)$. If all eight of the function values are greater than F_0 , the function value corresponding to the solution, then \mathbf{t}_0 is accepted as the global minimum. If at least one of the eight function values is smaller than F_0 then the Simplex is contracted around the new lowest point and the procedure is continued. A disadvantage of this method is that the criterion is dependent on the step size used to generate the initial Simplex. A better approach would be to use a step size that is a function of the size of the Simplex at convergence. Gill et al. (1981) suggested a restart with a Simplex of center \mathbf{t}_0 and side $\|\mathbf{t}_0 - \mathbf{t}_{n+1}\|$, the Euclidean distance between the best and worst point of the Simplex at initial convergence. Similarly, in each analysis $-\Lambda$ was evaluated at the eight points calculated from \mathbf{t}_0 plus and minus $\|\mathbf{t}_0 - \mathbf{t}_{n+1}\|$ in each direction. The test, which requires only $2n$

additional likelihood evaluations, was always correct in determining whether the estimate at initial convergence was a local or global minimum and should routinely be incorporated into SM to increase the confidence that the global minimum has been found.

The vector of estimates obtained for the Θ_{am} prior .10 did not converge to the global minimum even after restarting with a new Simplex but was identified as a local minimum by the method of Gill et al. (1981) described above. Compared to the analyses which converged to the global minimum on the first or restart run, the initial analyses for .10 and .20 were characterized by a relatively large percentage of rounds consisting of a failed contraction and subsequent shrinkage of the Simplex. This action occurred in 10 of 57 rounds with a Θ_{am} prior of .10 and in all 16 rounds with a prior of .20 but in at most one round for all other priors. Failed contractions may cause the Simplex to shrink into a valley and result in false convergence. Therefore, if the first run consists of several failed contractions, the best action is probably to restart the procedure with a set of priors different from those obtained at initial convergence. Because a failed contraction and shrinkage requires $n+2$ likelihood evaluations, the ratio of the number of likelihoods to the number of rounds of iteration will be large if there are several failed contractions. For these analyses, this ratio was 2.11 and 6.31 for priors of Θ_{am} of .10 and .20 but ranged from only 1.63 to 2.03 for the others (Table 1).

Another characteristic unique to the analyses with Θ_{am} priors of .10 and .20 was the large REML estimate for σ_p^2 relative to observed variance estimated from the total sum of squares. The latter estimate, calculated within each of the two generations, averaged 98.4. The estimate calculated as the sum of components from MT of REML should not in general be equal to this value, but it should be close. For the priors which converged to the global minimum initially or after a restart, $\hat{\sigma}_p^2$ ranged from 96.3 to 96.5, but the estimates for priors .10 and .20 were 105.0 and 176.4, respectively. Comparison of the variance estimates from the two methods may be helpful in determining when a new set of priors should be used and calculation of total sum of squares would be inexpensive for most analyses.

TABLE 1. Rounds of iteration and log likelihood evaluations required and parameter estimates by the Simplex method at initial convergence¹ using different prior values for Θ_{am}

Prior		Rounds	Likelihoods	-2Λ	Estimates					
Θ_{am}	r_{am}				$\hat{\sigma}_p^2$	$\hat{\Theta}_a$	$\hat{\Theta}_m$	$\hat{\Theta}_{am}$	$\hat{\Theta}_c$	\hat{r}_{am}
-.240	-.980	85	147	950.0604	96.4	.329	.156	.084	-.086	-.381
-.200	-.816	65	113	950.0604	96.3	.328	.158	.083	-.088	-.384
-.150	-.612	61	107	950.0604	96.3	.330	.157	.083	-.088	-.385
-.100	-.408	58	100	950.0604	96.4	.330	.157	.084	-.088	-.384
-.050	-.204	36	73	950.0617	96.3	.317	.139	.087	-.067	-.317
-.020	-.082	41	79	950.0712	96.3	.294	.105	.094	-.029	-.166
-.010	-.041	33	66	950.0776	96.3	.284	.094	.096	-.015	-.090
-.001	-.004	38	74	950.0846	96.3	.275	.082	.100	-.002	-.010
.001	.004	33	65	950.0865	96.4	.273	.081	.100	.002	.010
.010	.041	38	72	950.0952	96.5	.268	.066	.104	.015	.110
.020	.082	80	140	950.0604	96.3	.327	.156	.084	-.086	-.379
.050	.204	109	181	950.0604	96.3	.330	.158	.083	-.088	-.385
.100	.408	57	120	950.6663	105.0	.176	.071	.120	.111	.998
.150	.612	104	177	950.0604	96.3	.328	.157	.083	-.086	-.380
.200	.816	16	101	955.4507	176.4	.400	.150	.100	.200	.816
.240	.980	96	156	950.0618	96.4	.328	.142	.089	-.076	-.352

¹Convergence criterion: variance (-2Λ) < 1.e-9.

TABLE 2. Rounds of iteration and log likelihood evaluations required and parameter estimates by Powell's method at initial convergence¹ using different prior values for Θ_{am}

Prior		Rounds	Likelihoods	-2λ	Estimates					
Θ_{am}	r_{am}				$\hat{\sigma}_p^2$	$\hat{\Theta}_a$	$\hat{\Theta}_m$	$\hat{\Theta}_{am}$	$\hat{\Theta}_c$	\hat{r}_{am}
-.240	-.980	7	202	950.0604	96.3	.329	.157	-.087	.084	-.382
-.200	-.816	10	120	950.0604	96.3	.329	.157	-.087	.083	-.382
-.150	-.612	8	107	950.0604	96.4	.329	.157	-.087	.084	-.382
-.100	-.408	9	94	950.0604	96.3	.329	.157	-.087	.083	-.382
-.050	-.204	6	109	950.0604	96.3	.329	.157	-.087	.083	-.382
-.020	-.082	7	145	950.0604	96.3	.329	.157	-.087	.083	-.382
-.010	-.041	8	170	950.0604	96.4	.329	.157	-.087	.084	-.382
-.001	-.004	8	183	950.0604	96.4	.329	.157	-.087	.084	-.382
.001	.004	8	186	950.0604	96.4	.329	.157	-.087	.084	-.382
.010	.041	8	198	950.0604	96.3	.329	.157	-.087	.084	-.382
.020	.082	8	207	950.0604	96.4	.329	.157	-.087	.084	-.383
.050	.204	8	285	950.0604	96.3	.329	.157	-.087	.084	-.382
.100	.408	9	363	950.0604	96.3	.329	.157	-.087	.084	-.381
.150	.612	8	269	950.0604	96.3	.329	.157	-.087	.084	-.382
.200	.816	4	220	950.0629	96.2	.311	.131	-.059	.089	-.292
.240	.980	8	331	950.0604	96.3	.329	.157	-.087	.083	-.382

¹Convergence criterion: fractional change in all $\Theta < 1.e-3$.

TABLE 3. Rounds of iteration and log likelihood evaluations required and parameter estimates by the Simplex and Powell's method after restarting using priors obtained at initial convergence¹

Method	Initial prior		Rounds ²	Likelihoods ²	-2Λ	Estimates					
	Θ_{am}	r_{am}				$\hat{\Sigma}_p^2$	$\hat{\Theta}_a$	$\hat{\Theta}_m$	$\hat{\Theta}_{am}$	$\hat{\Theta}_c$	\hat{r}_{am}
Simplex	-.050	-.204	41 (77)	77 (150)	950.0604	96.3	.329	.157	-.087	.084	-.383
Simplex	-.020	-.082	57 (98)	110 (189)	950.0604	96.4	.328	.156	-.085	.084	-.378
Simplex	-.010	-.041	43 (76)	83 (149)	950.0605	96.3	.328	.155	-.085	.084	-.378
Simplex	-.001	-.004	63 (101)	109 (183)	950.0605	96.3	.329	.156	-.087	.084	-.384
Simplex	.001	.004	95 (128)	176 (241)	950.0605	96.4	.329	.159	-.088	.082	-.384
Simplex	.010	.041	66 (104)	111 (183)	950.0605	96.3	.328	.154	-.084	.084	-.374
Simplex	.100	.408	67 (124)	86 (206)	950.3779	100.5	.201	.062	.112	.075	1.000
Simplex	.240	.980	34 (130)	66 (222)	950.0605	96.3	.327	.156	-.085	.084	-.376
Powell's	.200	.816	7 (11)	78 (298)	950.0604	96.4	.329	.157	-.087	.084	-.382

¹Convergence criterion: Simplex variance (-2Λ) < 1.e-9; Powell's fractional change in all Θ < 1.e-3.

²Total for initial plus restart run in parentheses.

Powell's Method

In all but one of the analyses of the simulated data utilizing PM, parameter estimates obtained at convergence (Table 2) were invariant to the starting value selected for Θ_{am} and were identical to the estimates at the global minimum obtained with SM (Table 2). The Θ_{am} prior of .20 (which resulted in false convergence with SM) resulted in convergence to a local minimum, most likely as the result of premature termination of the method as indicated by the number of rounds of iteration. A linear direct search method which adds a conjugate search direction in each round will minimize a quadratic function of n independent variables in a minimum of $n+1$ rounds. Because PM does not always add a conjugate search direction in each round and Λ is not a quadratic function, more than $n+1=5$ rounds should be required for the example data. As expected, the number of rounds required to reach convergence ranged from six to ten for all Θ_{am} priors except for .20 which required only four (Table 2). Therefore, even though PM was not robust for all Θ_{am} priors, the failure to locate the minimum would be suspected from the number of rounds of iteration required. In this situation the analysis should be rerun with a different prior vector. If more than $n+1$ rounds are required then the estimates would be accepted as the those corresponding to the global minimum. Restarting the procedure with the initial estimates required seven rounds of iteration and 78 likelihood evaluations and resulted in convergence to the global minimum (last line of Table 3).

The different estimates obtained via SM and PM are probably due to the different convergence criteria used. Convergence for SM is based on differences in values of Λ while PM convergence is based on the fractional change of the parameter vector. Thus the first criterion is susceptible to premature termination at a plateau of Λ while the latter criterion can terminate prematurely on a very steep slope. As indicated by the small variation in -2Λ for the wide range of parameter estimates (Table 1), the surface of Λ for simulated data is relatively flat so SM stops at a local minimum rather than the global minimum, but PM is more robust. The degree of flatness of the log-likelihood surface is determined in part by the amount of information available which is dependent on the amount and structure of the data. In a model with correlated direct and maternal genetic effects, if few dams have records, the covariance component is estimated indirectly through the relationship matrix and the likelihood surface tends to be flat (Karin Meyer, personal communication, 1990). In the two generations of simulated data used for these analyses the flat likelihood surface is partly due to the fact that only 18 of 36 dams had records. Convergence to different estimates has been noted, however, in an analysis of data from a seven generation swine selection experiment in which all dams (643) except those in the base generation

had records (Luis Gama, personal communication, 1990). Limited investigation of analysis of the data for a model without a correlated maternal effect indicated that both SM and PM were robust for different Θ_a and Θ_c priors. The probability of false convergence is probably not as great for models in which Λ is maximized with respect to few parameters.

Efficiency

If alternative direct search procedures are robust then a second characteristic to consider is their efficiency. The computer time required to define a new vertex for a Simplex in SM or a new direction vector in PM is insignificant in comparison to the time required to calculate Λ via SPARSPAK. Therefore, the appropriate criterion is the number of likelihood evaluations required.

Simplex Method

For fixed reflection, expansion, and contraction coefficients the number of function evaluations required with SM is determined in part by the orientation and size of the original Simplex, i.e., the n coordinates of the starting point and the step size used to generate the initial Simplex. O'Neill (1971) stated that SM will work successfully for all step sizes and priors but those far from the optimum will require more iterations to reach convergence. Because the optimum step size and priors depend on the location of the function minimum, however, they are never known. As expected, when the prior for Θ_{am} was moved from -.10, a value close to the convergence point of -.09, to -.24, a point far away, the number of likelihood evaluations increased (Table 1). This trend was also evident for some of the positive priors which initially converged to the global minimum, e.g., .02 and .05, but a prior of .15 required fewer likelihood evaluations, 177, than the prior .05, which required 181. Meyer (1989) used a step size of .20 in her DFREML program to ensure convergence even for poor starting values but suggested that a smaller value might be more efficient for good starting values. Results from the analyses indicate that a smaller step size might be appropriate for restarts because for most priors which converged to the global minimum after a restart, the second run required more likelihood evaluations than the first and, as pointed out by Meyer (1989), a large step size may result in excessive likelihood evaluations.

Powell's Method

Similar to SM, PM should require fewer function evaluations for priors close to the converged estimate. This trend was observed over the entire range of Θ_{am} priors except for .20 which did not converge to the global minimum, and .05 and .10 which required more iterations than .15 (Table 2).

Modifications of PM to speed convergence could be directed at either the method in which

search directions are generated or at the linear search technique. The large variation in number of likelihoods and small variation in number of rounds of iteration (sets of direction vectors) suggests that the linear search using a quadratic approximation was the limiting component. In Powell's linear search step, the extrapolated step length, λ' , to the predicted minimum may be too long, i.e., far from the true minimum, resulting in unstable convergence. Therefore, Powell set the step length to a specified maximum permissible value when the predicted step is too long. For these analyses, this precautionary step resulted in very slow convergence of the linear search. Because the maximum step length is proportional to the accuracy criterion, one possible solution is to run the analysis twice, first with a relatively large accuracy criterion and then with a smaller value (Karin Meyer, personal communication, 1990). With this method, a large step size is used in the first run and a smaller value is used for more precise location of the minimum. For the example data, this two-stage method with an initial accuracy of 1.0 followed by 0.1 decreased the number of likelihood evaluations required for most priors, with a reduction of 50% or more observed for priors far from the converged estimates. An alternative approach suggested by Coggins as described by Box et al. (1969) is to double the step length until the minimum is bracketed in the first linear interpolation in each direction and then to use repeated quadratic interpolation. Jacoby et al. (1972) reported that this method is more efficient than Powell's linear search especially when the initial guess of the location of the minimum is poor.

Over the range of priors tested which converged to the global minimum initially or after a restart, neither SM nor PM was consistently more efficient. For priors far from the converged value ($<-.15$ and $>.001$) SM required fewer Λ evaluations than PM probably due to the slow convergence of Powell's linear search. For most intermediate priors, initial runs of SM were more efficient than PM, but evaluations required for restarts (Table 3) of the former resulted in a small advantage for PM.

Conclusions

Direct search strategies to locate the minimum of a function are an alternative to gradient-based methods and may be preferred for problems in which differentiation is difficult, e.g., REML estimation in AM. While both SM and PM are based on simple comparisons of function values, they use different strategies to generate a sequence of improved approximations to the solution. In contrast to PM which is based on conjugate directions, SM, having no mathematical basis, is more heuristic in nature so the former may be preferable on a theoretical basis.

Parameter estimates for the example data obtained at initial convergence with PM and

especially with SM were not independent of the prior used for Θ_{am} . Restarting the methods with the initial estimates resulted in the same final estimates in all PM and most SM analyses. Fortunately, evaluation of $-\Lambda$ in the vicinity of the estimates obtained with SM was able to differentiate between local solutions and global minima. In addition, if the progression of the methods is closely monitored then occurrences of false convergence will often be recognized. This emphasizes the importance of a basic understanding of the particular direct search method used rather than simply employing it as a 'black box'.

For the small data set and priors used, both methods were similar in terms of efficiency but many likelihood evaluations were often required to obtain final estimates. Parkinson and Hutchinson (1972) reported that SM is highly competitive with alternative direct search procedures especially for complex functions of low dimensionality, and therefore, the model used in these analyses should be well suited to SM. However, SM may become relatively less competitive as the number of dimensions are increased (Box, 1966). On the basis of limited work, Meyer (1990) suggested that PM may be more efficient than SM for REML in multivariate models especially for those with more than one random effect. In addition, like SM, PM can efficiently accommodate constraints on the parameter space which is important especially as the number of parameters increases.

In terms of reliability PM may be superior to SM for maximization of Λ with respect to variance components in animal models with correlated direct and maternal effects. If a linear quadratic approximation method is used in models with a single parameter (Graser, et al., 1987), then use of PM, which consists of a sequence of linear searches together with a method to generate search directions, would provide a unified approach to the maximization of Λ in DFREML.

APPENDIX: FSPAK Invoice

Product: Commercial distribution and maintenance for FSPAK - an interface to sparse matrix routines by George and Liu, with support for operations used in animal breeding including the sparse inversion.

Cost: \$400

To:
.....
.....
.....

From: Ignacy Misztal
University of Illinois, 1207 W Gregory Dr., Urbana IL 61801,USA
tel.(217) 244-3164, fax 333-8286, E-mail: ignacy@uiuc.edu

Please make the check to the University of Illinois and send it to Ignacy Misztal.

Disclaimer: FSPAK is available as-is. Before relying on it commercially, please verify that it operates correctly in a given application.

Voluntary distribution and maintenance fee is requested from commercial companies that use FSPAK. The fee is not strictly necessary to use FSPAK but it is essential to cover FSPAK's distribution and development costs. It ensures that FSPAK will be maintained and made available to operate on the current computer equipment with high reliability expected from commercial applications.

Bibliography

- Boldman, K. G. and L. D. Van Vleck. 1990. Effect of different starting values on parameter estimates by DF-REML and EM-REML in an animal model with maternal effects. *J. Anim. Sci.* 68(suppl. 2):71 (abstr.).
- Boldman, K. G. and L. D. Van Vleck. 1991. Derivative-free restricted maximum likelihood estimation in animal models with a sparse matrix solver. *J. Dairy Sci.* 74:4337.
- Box, M. J. 1966. A comparison of several current optimization methods, and the use of transformations in constrained problems. *Comp. J.* 8:42.
- Box, M. J., D. Davies and W. H. Swann. 1969. Non-linear optimization techniques. ICI Ltd., Monograph No. 5, Oliver and Boyd, Edinburgh.
- Chu, E., A. George, J. Liu, and E. Ng. 1984. SPARSPAK: Waterloo sparse matrix package user's guide for SPARSPAK-A. CS-84-36, Dept. Computer Sci., Univ. Waterloo, Waterloo, ON, Canada.
- Cunningham, E. P., and C. R. Henderson. 1968. An iterative procedure for estimating fixed effects and variance components in mixed model situations. *Biometrics* 24:13.
- George, A., and J. W. H. Liu. 1980. *Computer solution of large sparse positive definite systems.* Prentice-Hall, Englewood Cliffs, NJ.
- George, A., J. Liu, and E. Ng. 1980. User guide for SPARSPAK: Waterloo sparse linear equations package. CS-78-30, Dept. Computer Sci., Univ. Waterloo, ON, Canada.
- George, A., and E. Ng. 1984. A new release of SPARSPAK: The Waterloo sparse matrix package. Mimeo, Dept. Computer Sci., Univ. Waterloo, Waterloo, ON, Canada.
- Gill, P. E., W. Murray, and M. H. Wright. 1981. *Practical Optimization.* Academic Press, New York.
- Graser, H. -U., S. P. Smith, and B. Tier. 1987. A derivative-free approach for estimating variance components in animal models by restricted maximum likelihood. *J. Anim. Sci.* 64:1362-1370.
- Groeneveld, E. and M. Kovac. 1990. A note on multiple solutions in multivariate restricted maximum likelihood covariance component estimation. *J. Dairy Sci.* 73:2221.
- Harville, D. A. 1977. Maximum likelihood approaches to variance component estimation and to related problems. *J. Amer. Stat. Assoc.* 72:320-340.
- Harville, D. A. 1979. Some useful representations for constrained mixed-model estimation. *J. Am. Stat. Assn.* 74:200.
- Henderson, C. R. 1950. Estimation of genetic parameters. *Ann. Math. Stat.* 21:309.
- Henderson, C. R. 1963. Selection index and expected genetic advance. In: *Statistical Genetics in Plant Breeding.* NAS-NRC publication 982.
- Henderson, C. R. 1974. General flexibility of linear model techniques for sire evaluation. *J. Dairy Sci.* 57:963.
- Henderson, C. R. 1975. Best linear unbiased prediction under a selection model. *Biometrics* 31:423.
- Henderson, C. R. 1984a. *Application of linear models in animal breeding.* U. Guelph, Guelph,

ON, Canada.

- Henderson, C. R., O. Kempthorne, S. R. Searle, and C. N. von Krosigk. 1959. Estimation of environmental and genetic trends from records subject to culling. *Biometrics* 15:192.
- Jacoby, S. L. S., J. S. Kowalik and J. T. Pizzo. 1972. *Iterative methods for nonlinear optimization problems*. Prentice-Hall, Englewood Cliffs, NJ.
- Kovac, M. and E. Groeneveld. 1989. Local maxima in multiple trait variance component estimation. *J. Anim. Sci.* 67(suppl. 1):33.
- Kuester, J. L. and J. H. Mize. 1973. *Optimization techniques with FORTRAN*. McGraw-Hill, New York.
- Meyer, K. 1985. Maximum likelihood estimation of variance components for a multivariate mixed model with equal design matrices. *Biometrics* 41:153.
- Meyer, K. 1988a. DFREML. A set of programs to estimate variance components under individual animal model. *J. Dairy Sci.* 71(suppl. 2):33-34.
- Meyer, K. 1988b. DFREML. Programs to estimate variance components for individual animal models by restricted maximum likelihood. User notes. Univ. of Edinburgh.
- Meyer, K. 1989. Restricted maximum likelihood to estimate variance components for animal models with several random effects using a derivative-free algorithm. *Genet. Sel. Evol.* 21:317-340.
- Meyer, K. 1991. Estimating variances and covariances for multivariate animal models by restricted maximum likelihood. *Genet. Sel. Evol.* 23:67.
- Misztal, I. 1990. Restricted maximum likelihood estimation of variance components in an animal model using sparse matrix inversion and a supercomputer. *J. Dairy Sci.* 73:163.
- Nelder, J. A. and R. Mead. 1965. A simplex method for function minimization. *Computer J.* 7:308.
- O'Neill, R. 1971. Algorithm AS 47: Function minimization using a simplex procedure. *Appl. Stat.* 20:338-345.
- Parkinson, J. M. and D. Hutchinson. 1972. A consideration of non-gradient algorithms for the unconstrained optimization of functions of high dimensionality. Page 99 *in* Numerical methods for non-linear optimization. F. A. Lootsma ed. Academic Press, New York.
- Patterson, H. D. and R. Thompson. 1971. Recovery of inter-block information when block sizes are unequal. *Biometrika* 58:545.
- Pollak, E. J., and R. L. Quaas. 1982. Alternative strategy for building multiple trait mixed model equations. *J. Dairy Sci.* 68(Suppl.1):102.
- Powell, M. J. D. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer J.* 7:155.
- Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 1989. *Numerical recipes*. Cambridge University Press, New York.
- Quaas, R. L. 1976. Computing the diagonal elements and inverse of a large numerator relationship matrix. *Biometrics* 32:949.
- Searle, S. R. 1979. Notes on variance component estimation: A detailed account of maximum

- likelihood and kindred methodology. Paper BU-673-M, Biometrics Unit, Cornell Univ.
- Searle, S. R. 1982. Matrix algebra useful for statistics. John Wiley and Sons, New York.
- Smith, S. P. and H. -U. Graser. 1986. Estimating variance components in a class of mixed models by restricted maximum likelihood. *J. Dairy Sci.* 69:1156-1165.
- Stewart, G. W. 1973. *Introduction to matrix computations*. Academic Press, New York.
- Thompson, R. 1969. Iterative estimation of variance components for non-orthogonal data. *Biometrics* 25:767.
- Tier, B. and S. P. Smith. 1989. Use of sparse matrix absorption in animal breeding. *Genet. Sel. Evol.* 21:457-466.
- Walter, J. P., I. L. Mao, and B. D. Banks. 1986. An iterative algorithm for estimation of variance and covariance components in a multiple trait model using triangular decomposition of residual variance - covariance matrix and absorption of fixed effects. *J. Dairy Sci.* 69(Suppl.2):207.