COMPUTING SOLUTIONS TO MIXED MODEL EQUATIONS

L.R. SCHAEFFER and B.W. KENNEDY, CANADA

Centre for Genetic Improvement of Livestock
Department of Animal & Poultry Science
University of Guelph
Guelph, Ontario N1G 2W1, Canada

SUMMARY

A simple method for solving mixed model equations of large order is presented for single trait models. A comparison of the simple method to a modified reduced animal model procedure for swine indicated the simple method was less time consuming and converged faster than the reduced animal model under the criterion of .5% maximum change in solutions relative to the standard deviation of solutions. However, convergence at a .1% criterion convergence was dependent on the trait being analyzed. Extension of the simple method to multiple trait models is presented, as well as a procedure for estimating variance-covariance matrices by maximum likelihood using a Cholesky decomposition transformation.

INTRODUCTION

The process of constructing and solving the mixed model equations of Henderson (1973) for genetic evaluation of livestock can be both costly and complex. Attempts to compare and improve upon various computational strategies are found readily in the literature (Ufford et al., 1979; Quaas and Pollak, 1981; Blair and Pollak, 1984; Hudson, 1984; Schaeffer and Kennedy, 1985; and Van Vleck and Dwyer, 1985). Comparisons include the number of programs and relative computing times of each, and the number of iterations needed to reach a specified criterion of convergence. Another means of comparing algorithms could be according to the types of models that may be handled. Most computer programs are specific to one model or one class of models, such as the reduced animal model.

Usually the number of equations is too large to permit an explicit solution or the use of SAS or programs of Harvey (1975), and solutions must be obtained by iterative procedures. In such cases, the use of REML (Restricted Maximum Likelihood) or MIVQUE (Minimum Variance Quadratic Unbiased Estimation) for estimating variance components is essentially impossible even for single trait analyses. ML (Maximum Likelihood), however, may be possible with certain models and Smith and Graser (1985) present a REML procedure for a particular class of models.

The objectives of this paper are to:

1. describe a simple procedure for solving mixed model equations without constructing the equations explicitly,
2. compare the simple procedure to a reduced animal model procedure as applied to Canadian swine data,
3. extend the simple procedure for multiple traits, and
4. indicate how variances and covariances may be estimated by ML.

THE SIMPLE PROCEDURE

To describe this computing algorithm in completely general terms for any model would be cumbersome. Instead we will describe the algorithm for one class of models and leave the reader to extrapolate to other classes of

models. A small example is included. Let the model equation be

$$y = Wa + Xb + Zc + e \qquad [1]$$

where

- **a** is a vector of fixed effects, of length p and p is small enough to allow **W'W** to be stored in memory;
- **b** is a vector of either fixed or random effects of length t and t is usually a very large number and **X'X** is diagonal;
- **c** is a vector of random effects which can be either sire or animal effects;
- **e** is a residual vector;
- **X** and **Z** are design matrices of zeros or ones, but **W** may include covariates and more than one fixed factor so that **W'W** is not necessarily diagonal.

The expectations and variance-covariance matrices for fixed **b** are

$$E\begin{bmatrix} y \\ c \\ e \end{bmatrix} = \begin{bmatrix} Wa + Xb \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad V\begin{bmatrix} c \\ e \end{bmatrix} = \begin{bmatrix} A\sigma_c^2 & 0 \\ 0 & I\sigma_e^2 \end{bmatrix}$$

where **I** is an identity matrix and **A** is the additive genetic numerator relationship matrix. Let

$$Zc = (Z_1 \quad 0)\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

where c_1 represents sires with progeny or animals with records in **y**, and
c_2 represents relatives of individuals in c_1 that are needed to compute **A** and which are not included in **y**. The inverse of **A** will be calculated by Henderson's (1975) rules assuming non-inbred individuals in **c**.

Consider the example data in Table 1 in which age group effects belong to **a**, herd-year-season effects belong to **b** and sires belong to c_1. The pedigrees of the sires appear in Table 2. Assuming that the ratio of residual to sire variances is 11, the mixed model equations and solutions are given in Table 3. We now describe how to obtain the same solutions without constructing the equations.

Firstly, the data must be prepared such that the levels of each factor (fixed or random) are numbered consecutively. This includes individuals in c_2. Secondly, for this model, we need one copy of the data sorted by levels of **b** and another copy sorted by levels of **c**. Lastly, we need to construct a coded pedigree file to facilitate calculation of elements of A^{-1} as we need them. The coded pedigree file is created as follows: let i=bull, j=sire, and k=maternal grandsire, then write the following data to the coded pedigree file:

(i, 1, j, k)
(j, 2, i, k)
(k, 3, i, j).

Table 1. Example data

Age Group	Herd-Year-Season	Sire	Observation
1	1	1	704
1	1	1	285
1	1	2	1495
1	1	2	1117
2	1	2	1106
2	1	3	1198
2	2	1	837
1	2	1	576
1	2	1	412
3	2	2	529
1	2	3	1441
3	3	1	624
1	3	2	781
2	3	2	846
1	3	3	1605

Table 2. Pedigrees of sires in example

Bull	Sire	MGS
1	5	4
2	6	4
3	1	6
4	-	-
5	-	-
6	-	-

Table 3. Mixed model equations for example data

$$\begin{pmatrix} 9 & 0 & 0 & 4 & 3 & 2 & 4 & 3 & 2 & 0 & 0 & 0 \\ 0 & 4 & 0 & 2 & 1 & 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 4 & 2 & 0 & 6 & 0 & 0 & 2 & 3 & 1 & 0 & 0 & 0 \\ 3 & 1 & 1 & 0 & 5 & 0 & 3 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 & 4 & 1 & 2 & 1 & 0 & 0 & 0 \\ 4 & 1 & 1 & 2 & 3 & 1 & 26 & 0 & -8 & -4 & -8 & 2 \\ 3 & 2 & 1 & 3 & 1 & 2 & 0 & 22 & 0 & -4 & 0 & -8 \\ 2 & 1 & 0 & 1 & 1 & 1 & -8 & 0 & 19 & 0 & 0 & -4 \\ 0 & 0 & 0 & 0 & 0 & 0 & -4 & -4 & 0 & 13 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & -8 & 0 & 0 & 2 & 15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -8 & -4 & 2 & 0 & 16 \end{pmatrix} \begin{pmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \\ \hat{c}_1 \\ \hat{c}_2 \\ \hat{c}_3 \\ \hat{c}_4 \\ \hat{c}_5 \\ \hat{c}_6 \end{pmatrix} = \begin{pmatrix} 8416 \\ 3987 \\ 1153 \\ 5905 \\ 3795 \\ 3856 \\ 3438 \\ 5874 \\ 4244 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$\hat{a}_1 = 0$ $\quad \hat{b}_1 = 976.5164$ $\quad \hat{c}_1 = -68.7260$ $\quad \hat{c}_4 = -13.6979$

$\hat{a}_2 = 30.5239$ $\quad \hat{b}_2 = 847.8510$ $\quad \hat{c}_2 = 24.2079$ $\quad \hat{c}_5 = -34.8275$

$\hat{a}_3 = -342.4908$ $\quad \hat{b}_3 = 1034.6487$ $\quad \hat{c}_3 = 49.6822$ $\quad \hat{c}_6 = 34.8275$

If j is zero, omit code 2 data, and if K is zero, omit code 3 data, but all elements of c should have a code 1 record. Then sort the coded pedigree file according to the first two numbers in ascending order. The sorted, coded pedigree file for the example data is given in Table 4. The coded pedigree file allows all male progeny and maternal grand progeny of a particular individual to be grouped together.

Table 4. Coded pedigree file for example data

Individual	Code	j(i)	K(j)
1	1	5	4
1	2	3	6
2	1	6	4
3	1	1	6
4	1	0	0
4	3	1	5
4	3	2	6
5	1	0	0
5	2	1	4
6	1	0	0
6	2	2	4
6	3	3	1

The solution program consists of four sections. The first section involves the definition and initialization of arrays. Let WW represent the storage array for $\mathbf{W'W}$, WY for $\mathbf{W'y}$, AS for $\hat{\mathbf{a}}$, CS for $\hat{\mathbf{c}}$, and AC and CC are work vectors for $\hat{\mathbf{a}}$ and $\hat{\mathbf{c}}$, respectively. The scalar variables BS and BC refer to a particular element of $\hat{\mathbf{b}}$. Hence, the program requires enough computer memory to hold AS, CS, AC, CC, WW and WY. If the number of elements of $\hat{\mathbf{a}}$ is 150, then WW requires a maximum of 90,600 bytes (assuming all variables are REAL*8), AS, AC, and WY require another 3,600 bytes or a total of 92K. If the number of sires is 35,000, then CS and CC require another 547K. Most main frame computers today allow the use of 8 Megabytes or more, which would enable 512,000 sires to be evaluated.

Initial solutions for $\hat{\mathbf{a}}$, $\hat{\mathbf{b}}$ and $\hat{\mathbf{c}}$ are zero. $\mathbf{W'W}$ and $\mathbf{W'y}$ are stored into WW and WY arrays, and restrictions on solutions, $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$, should be imposed and an inverse of the restricted $\mathbf{W'W}$ computed and stored in WW. In the example, the first level of $\hat{\mathbf{a}}$ will be made zero.

The second section of the program reads the data sorted by levels of factor **b**. For each record within a level of **b** do the following calculations:

1. Accumulate (Y-AS(IA)-CS(IC)) in BC
2. Count the number of records, NB
3. Keep track of the number of times that each level of **a** occurs within a level of **b**, BA(IA) = BA(IA)+1
4. Accumulate (Y-AS(IA)) into CC(IC), and
5. Keep track of the number of times that each level of **c** occurs within a level of **b**.

IA, IB, and IC are the level identifiers for factors a, b, and c, respectively, and Y is the observation on the trait of interest.

For the first herd-year-season in the example, we would obtain

$$BC = 5905, \quad NB = 6,$$

$$BA = \begin{bmatrix} 4 \\ 2 \\ 0 \end{bmatrix}, \text{ and } CC = \begin{bmatrix} 989 \\ 3718 \\ 1198 \end{bmatrix}.$$

The new solution for the first herd-year-season becomes

$$BS = BC/NB = 984.1667.$$

Before proceeding to the next herd-year-season, the right hand sides for factors a and c can be adjusted for this new solution.

$$CC = CC - \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} BS = \begin{bmatrix} -979.3333 \\ 765.5 \\ 213.8333 \end{bmatrix}$$

and

$$AC = AC - \begin{bmatrix} 4 \\ 2 \\ 0 \end{bmatrix} BS = \begin{bmatrix} -3936.6667 \\ -1968.3333 \\ 0 \end{bmatrix}.$$

At this point, BS is no longer needed and the next herd-year-season may be processed. For the next herd-year-season in the example we find

$$BS_2 = 3795/5 = 759$$

$$CC = \begin{bmatrix} -979.3333 \\ 765.5 \\ 213.8333 \end{bmatrix} + \begin{bmatrix} 1825 \\ 529 \\ 1441 \end{bmatrix} - \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} BS_2 = \begin{bmatrix} -1431.3333 \\ 535.5 \\ 895.8333 \end{bmatrix}$$

and

$$AC = \begin{bmatrix} -3936.6667 \\ -1968.3333 \\ 0 \end{bmatrix} - \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} BS_2 = \begin{bmatrix} -6213.6667 \\ -2727.3333 \\ -759.0 \end{bmatrix},$$

and for the last herd-year-season,

$$BS_3 = 3856/4 = 964$$

$$CC = \begin{bmatrix} -1431.3333 \\ 535.5 \\ 895.8333 \end{bmatrix} + \begin{bmatrix} 624 \\ 1627 \\ 1605 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} BS_3 = \begin{bmatrix} -1771.3333 \\ 234.5 \\ 1536.8333 \end{bmatrix},$$

and $$AC = \begin{bmatrix} -6213.6667 \\ -2727.3333 \\ -759.0 \end{bmatrix} - \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} BS_3 = \begin{bmatrix} -8141.6667 \\ -3691.3333 \\ -1723.0 \end{bmatrix}.$$

The third section of the program reads both the coded pedigree file and the data sorted by levels of factor c (sires). For the first sire we find that its sire and maternal grandsire (MGS) are 5 and 4, respectively, and that sire 1 was the sire of sire 3, whose MGS was 6. Following Henderson (1975), we adjust the element of CC(1) as follows

$$CC(1) = CC(1) + (8/11)\,KC\,(CS(5) + 1/2\,CS(4)) + (8/11)\,KC\,(CS(3) - 1/4\,CS(6)),$$

where KC = 11 is the ratio of residual to sire variances, and we accumulate

$$(16/11)\,KC + (4/11)\,KC \text{ into CD}$$

where CD corresponds to the diagonal of A^{-1} for sire 1. From the data file we find that sire 1 has 6 progeny which we add into CD and we keep track of the number of times each level of a occurs with this sire. For the first sire, then

$$CC(1) = -1771.3333$$

$$CD = 6 + 16 + 4 = 26$$

or

$$CS(1) = -1771.3333/26 = -16.1282.$$

Before processing the next sire, adjust the right hand sides of factor a for the new sire solution

$$AC = AC - \begin{pmatrix} 4 \\ 1 \\ 1 \end{pmatrix} CS_1 = \begin{pmatrix} -7869.1539 \\ -3623.2051 \\ -1654.8718 \end{pmatrix}.$$

The remaining five sires are processed in a similar fashion.

The last section of the program calculates a new $\hat{a}$ by

$$AS = (W'W)^{-}(WY + AC).$$

Then the next iteration is begun going back to section two of the program.

The program may include a section to force $1'A^{-1}\hat{c}$ to be zero, and this may help to speed convergence. This program also facilitates the partitioning of sire proofs according to factors in the model as described by Schaeffer (1983). Such a program will be faster than one which constructs the equations explicitly if the number of non-zero coefficients in the equations is greater than twice the number of records in the data (for a fixed number of iterations).

APPLICATION

A national genetic evaluation system for swine for growth rate and backfat was introduced recently in Canada (Hudson and Kennedy, 1985; Kennedy and Hudson, 1985). The model contains fixed herd-year-season effects and random litter, animal (additive genetic) and residual effects, and computations of breeding values are according to a modified reduced animal model (RAM) as described in detail by Hudson and Kennedy (1985). The simple

method described here was also applied to the Canadian swine data and comparisons were made with the RAM computing procedure. For this application, **a**, **b** and **c** of equation [1] represent herd-year-season, litter and animal effects respectively. Comparisons between the RAM and simple computing methods are provided here only for the Yorkshire breed in Ontario, the largest data set analyzed. Specifics on numbers of records, equations and solutions are in Table 5.

Table 5. Data used for genetic evaluation of Yorkshire pigs in Ontario and numbers of equations, stored coefficients, records and solutions for modified reduced animal model (RAM) and simple computing methods.

	Number
Data	
Animals with records	86,385
Ancestors without records	6,821
Litters	25,777
Herd-year-seasons	1,733
RAM	
Equations	23,646
Stored coefficients	1,342,494
Parent solutions	14,349
Progeny solutions	27,442
Simple Method	
Pedigree file	265,976
Animal solutions	93,206

With RAM, 23,646 equations were created which resulted in 1,342,494 coefficients to be read each round of iteration. Only non-zero elements of the coefficient matrix were (full) stored. The simple method required reading a pedigree file of 265,976 records once and the observation file of 86,385 records twice each round of iteration, which was about one-third of the amount of read operations as with RAM. A total of 120 rounds of iteration were performed with each method. Both methods were computationally demanding. RAM required 84.2 minutes and the simple method required 36.3 minutes of CPU time to set up and solve the equations for both traits, age to 90 kg and backfat, on an IBM 3081 computer. Total computing time with the simple method was 43% of RAM, largely because of the need to read fewer records with each round of iteration. The RAM and simple method required 590 and 1032K of memory, respectively. Savings in computer time were similar for comparisons of analyses of two other breeds tested, but not reported here.

Solutions for 14,349 sires and dams were generated with RAM and evaluations of 27,442 recent progeny were then obtained by backsolving. The simple method provided evaluations on 93,206 animals, parents and their progeny, directly. Correlations between evaluations from the two methods were greater than 0.999 for both traits.

Rates of convergence differed for the three methods. Two criteria of convergence were compared - average absolute change and maximum change in animal solutions between rounds. The simple method converged more rapidly

initially than RAM, but after a number of rounds the rate of convergence of the simple method was less than RAM. As a result, the simple method and RAM were at equal levels of convergence by round 81 for average absolute change and round 97 for maximum change for backfat. Prior to these points, the simple method was at a higher degree of convergence and after the relative degree of convergence was greater for RAM. For days to 90 kg, however, RAM had not reached the same degree of convergence as the simple method by round 120, when iteration was stopped. Relative rates of convergence were trait dependent for both methods even though design matrices were identical for both traits.

Table 6 gives the number of rounds of iteration for each method to reach specific levels of convergence of less than 1.0, 0.5 and 0.1% change from the previous round. Change for each trait was measured relative to one standard deviation in animal solutions (4.65 days for age to 90 kg and 1.1 mm for backfat). RAM took 115 rounds for days to 90 kg and 79 rounds for backfat to reach a maximum change of less than 1% compared with 56 and 49 rounds for days to 90 kg and backfat with the simple method. To reach an average absolute change of less than 0.5% for days to 90 kg and backfat took 96 and 63 rounds with RAM and 20 and 21 rounds with the simple method. Both methods took about the same number of rounds of iteration to reach a very stringent convergence requirement of less than 0.1% average absolute change (81 and 83), but for days to 90 kg the simple method took only 58 rounds compared to 120 rounds with RAM. Computed changes in solutions were based on only 14,349 parent solutions with RAM but involved all 93,206 animal solutions with the simple method. Also, the RAM program had a restriction forcing $\mathbf{1}'\mathbf{A}^{-1}\hat{\mathbf{c}} = 0$ built in to speed convergence which the program for the simple method did not have, although it would be possible to add this restriction.

Table 6. Number of rounds of iteration required by the modified reduced animal model (RAM) and the simple method to reach convergence as measured by average absolute change and maximum change in animal solutions (expressed as percentage relative to one standard deviation of animal solutions).

Standardized percentage change	Average absolute change				Maximum change			
	Days to 90 kg		Backfat		Days to 90 kg		Backfat	
	RAM	Simple	RAM	Simple	RAM	Simple	RAM	Simple
1.0%	85	13	56	12	115	56	79	49
0.5%	96	20	63	21	>120	76	88	72
0.1%	120	58	81	83	>120	>120	108	>120

Based on these results and similar results with the other data sets examined, it was decided to replace the RAM method with the simple method for genetic evaluation of swine in Canada. In addition to the savings in computer time as a result of fewer read operations and rounds of iteration, the simple method was simpler operationally and allowed the consolidation of more than a dozen programs required for the RAM method into one program with the simple method.

EXTENSION TO MULTIPLE TRAITS

The simple procedure may be applied to multiple trait models, but the amount of memory may limit either the number of animals to be evaluated or the number of traits that can be included. To illustrate, consider a three-trait model where

$$y = Wa + Xb + Zc + e$$

where

a, b	are vectors of fixed effects,
c	is a vector of random sire effects,
e	is a vector of residual effects, and
W, X and **Z**	are design matrices.

Now we have,

$$V(c) = G*A \text{ and } G = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{12} & g_{22} & g_{23} \\ g_{13} & g_{23} & g_{33} \end{bmatrix}$$

where g_{ij} is the sire covariance between traits i and j, and **A** is the numerator additive genetic relationship matrix. Similarly, $V(e) = R$. If traits are ordered within animals, then **R** would be a block diagonal matrix. Let

$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{12} & e_{22} & e_{23} \\ e_{13} & e_{23} & e_{33} \end{bmatrix},$$

with e_{ij} being the covariance of residual effects between traits i and j measured on the same animal, and let

$$E_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & e_{22} & e_{23} \\ 0 & e_{23} & e_{33} \end{bmatrix}$$

if trait 1 were missing on an animal, and so on. With 3 traits there are seven $(2^t - 1)$ different possible **E**-matrices. The matrix **R** is the direct sum of the appropriate **E**-matrix for each animal. For the multiple trait version of the simple program we need to save the inverse of each of the seven possible **E**-matrices. If certain combinations of missing traits do not occur in the data, then there would be fewer **E**-matrices to retain.

The data should be sorted by levels of **b** and another copy by levels of **c**, and all levels of each factor consecutively numbered. The coded pedigree file is the same as before. $W'R^{-1}W$ and $W'R^{-1}y$ are stored in program memory. Initial solutions for all factors are zero.

Now read the data sorted by levels of **b**. For each animal the following calculations are performed:

1. Accumulate

   $$E_k^{-1}(y_k - \hat{a}_j - \hat{c}_\ell) \text{ into BC}$$

   where E_k^{-1} is the appropriate E^{-1} matrix for the kth individual, y_k is the txl observation vector (t-traits), $\hat{a}_j$ is the current solution for the jth a-effects associated with y_k, and $\hat{c}_\ell$ is the current solution for the ℓth sire of the kth animal.

2. Accumulate E_k^{-1} into BD

3. Accumulate $E_k^{-1}(y_k - \hat{a}_j)$ into CC(ℓ)

4. Keep track of levels of **a** and **c** that occur in this level of **b** and also which E^{-1} was associated with each occurrence.

After processing all animals in that level of **b**, compute the new solution as

$$BS = (BD)^{-1}(BC).$$

Since BD is the same each iteration one might save time by storing each $(BD)^{-1}$ on a temporary work disk and reading them back each time they are needed.

Now adjust AC and CC for BS as in the single trait procedure, before proceeding to the next level of **b**.

The next section of a program would read the data and the coded pedigree file for levels of **c** (sires). Suppose a bull (i) has sire j and MGS k, then accumulate

$$8/11\ G^{-1}(\hat{c}_j + 1/2\ \hat{c}_k) \text{ into CC(i)}$$

where $\hat{c}_j$ and $\hat{c}_k$ are the current solutions for sires j and k for t-traits, and accumulate

$$16/11\ G^{-1} \text{ into CD(i).}$$

From the data, accumulate the appropriate E^{-1} into CD(i) and keep track of the levels of **a** that occur in the ith level of **c** as well as the corresponding E^{-1}. The new solution for the ith sire is

$$\hat{c}_i = [CD(i)]^{-1}CC(i).$$

Again, $[CD(i)]^{-1}$ may be saved on a temporary file and retrieved sequentially as needed.

The last step is to compute a new $\hat{a}$ and repeat the process until convergence is achieved. Hence, the extension to multiple traits is not complicated.

ESTIMATION OF VARIANCES AND COVARIANCES

If we can assume the same model for each trait in a multi-trait model,

and assume that missing observations are due to sequential culling, then a Cholesky decomposition of $\mathbf{E}$ may be used to transform y so that $\mathbf{R} = \mathbf{I}$. By sequential culling we mean that if trait j is missing then traits 1 to $(j-1)$ must be present and traits j to t must be missing. Examples are milk yields in successive lactations, or body weights at various ages. The Cholesky decomposition of $\mathbf{E}$ is a lower triangular matrix, $\mathbf{T}$, such that

$$\mathbf{E} = \mathbf{TT}' \text{ and } \mathbf{T}^{-1}\mathbf{E}\mathbf{T}^{-1}{}' = \mathbf{I}.$$

Then $\mathbf{T}^{-1}y_{-1}$ is used rather than y. Parts of the multi-trait program that involved $\mathbf{E}^{-1}$ now only use $\mathbf{I}$, and in place of $\mathbf{G}^{-1}$ use

$$(\mathbf{T}^{-1}\mathbf{G}\mathbf{T}^{-1}{}')^{-1} = \mathbf{T}'\mathbf{G}^{-1}\mathbf{T} = \mathbf{G}_*^{-1}.$$

The solutions that are obtained are for $\mathbf{T}^{-1}\hat{a}$, $\mathbf{T}^{-1}\hat{b}$, and $\mathbf{T}^{-1}\hat{c}$, and these can be converted back to the original scale by premultiplying by $\mathbf{T}$. Let $\hat{c}_{*k}$ represent the solutions, on the transformed traits, for the kth sire. Recall that CD(k) was the submatrix for the kth level of $\mathbf{c}$. A maximum likelihood (ML) estimator of $\mathbf{G}_*$ (assuming $\mathbf{A}=\mathbf{I}$) would be

$$\hat{\mathbf{G}}_* = \left(\sum_{k=1}^{NC} (\hat{c}_{*k}\hat{c}_{*k}' + [CD(k)]^{-1}) \right) /NC$$

where NC is the number of the levels of $\mathbf{c}$. Then an estimate of $\mathbf{G}$ is

$$\hat{\mathbf{G}} = \mathbf{T}\hat{\mathbf{G}}_*\mathbf{T}'.$$

For the residual components, calculate for each transformed trait, the total sum of squares minus the solutions ($\hat{a}_*$, $\hat{b}_*$, and $\hat{c}_*$) times their corresponding right hand sides and divide by the number of observations for that trait less the rank of the fixed effects. This is a REML estimator of the residual variance rather than an ML estimator, and hence, the procedure is a combination of the two methods. These estimates should be close to unity if the prior values in $\mathbf{E}$ were correct. Let γ_i be the estimate of the residual variance for transformed trait i. Let $\mathbf{D}$ be a diagonal matrix with diagonals equal to γ_i, then

$$\hat{\mathbf{E}} = \mathbf{TDT}'.$$

Complete details of this method are given by Schaeffer (1985). Using the new $\hat{\mathbf{E}}$ and $\hat{\mathbf{G}}$, a new $\mathbf{T}$ can be obtained and the whole process repeated.

ACKNOWLEDGEMENT

The computing assistance of Margaret Quinton is gratefully acknowledged.

REFERENCES

Blair, H.T. and E.J. Pollak. 1984. Comparison of an animal model and an equivalent reduced animal model for computational efficiency using mixed model methodology. J. Anim. Sci. 58:1090-1096.

Harvey, W.R. 1975. Least squares analysis of data. U.S. Dept. Agriculture, ARS-H-4, Washington, D.C.

Henderson, C.R. 1975. Inverse of a matrix of relationships due to sires ad maternal grandsires. J. Dairy Si. 58:1917-1924.

Hudson, G.F.S. 1984. Extension of a reduced animal model to recursive prediction of breeding values. J. Anim. Sci. 59:1164-1175.

Hudson, G.F.S. and B.W. Kennedy. 1985. Genetic evaluation of swine for growth rate and backfat thickness. J. Anim. Sci. 61:83-91.

Kennedy, B.W. and G.F.S. Hudson. 1985. Application of an animal model to best linear unbiased prediction of breeding values of pigs. 36th Annual Meeting Europ. Assoc. Anim. Prod., Kallithea, Halkidiki, Greece, 30 Sept. - 3 Oct., 1985.

Quaas, R.L. and E.J. Pollak. 1981. Modified equations for sire models with groups. J. Dairy Sci. 64:1868-1872.

Schaeffer, L.R. 1983. Techniques for partitioning sire evaluations. J. Dairy Sci. 66:1519-1527.

Schaeffer, L.R. 1985. Estimation of variances and covariances within the allowable parameter space. J. Dairy Sci. (In Press).

Schaeffer, L.R. and B.W. Kennedy. 1985. Computing strategies for solving mixed model equations. J. Dairy Sci. (In Press).

Smith, S.P. and H.U. Graser. 1986. Estimating variance components in a class of models by restricted maximum likelihood. J. Dairy Sci. (Submitted).

Ufford, G.R., C.R. Henderson and L.D. Van Vleck. 1979. Computing algorithms for sire evaluation with all lactation records and natural service sires. J. Dairy Sci. 62:511-519.

Van Vleck, L.D. and D.J. Dwyer. 1985. Comparison of iterative procedures for solving equations for sire evaluation. J. Dairy Sci. 68:1006-1014.