

SOFTWARE PACKAGES IN ANIMAL BREEDING

Ignacy Misztal

University of Illinois, Urbana, IL 61801, USA; E-mail: ignacy@uiuc.edu

SUMMARY

Six packages for solving mixed model equations and estimating variance components by REML are compared. Each has a unique strength. Meticulously finished DFREML is the only one that implements hypothesis testing for variance component estimates. MTDFREML is a comprehensive, relatively easy to install and simple to use package for analyses of experimental data. PEST, the only semi-commercial program compared, has the best user interface and is the best supported program. The JAA/MTC package is the simplest and can compute multitrait estimates of variance components for many traits in the repeatability model. ABTK introduces tools that with built-in Unix utilities offer great flexibility to experienced users. Finally, DMU is a comprehensive attempt for a very general package, efficient for both small and large data sets, and with unique support for a Newton-Raphson type REML.

INTRODUCTION

More and more public software is available in animal breeding. To a large extent this is caused by demand for software and by availability of computer networks. Computing problems in animal breeding are becoming increasingly more complex, and fewer people are able to allocate large resources to develop sophisticated software. Computer networks can provide rapid distribution of almost any existing software across the globe in a matter of minutes. The same network provides electronic mail capability to support fast communication between software developers and users.

Sharing software has both advantages and disadvantages for software developers. On one hand, software is closely scrutinized and its bugs are more easily detected; also, developers receive wider recognition for their work. On the other hand, developers are expected to support their software for diverse applications on many computing platforms, and such a support may be very time-consuming.

Increasing number of available software packages with similar purposes calls for comparisons. It is difficult to select the best package for everyone because no package will meet everyone's objectives. First, requirements differ between projects and scientists. Execution speed rankings vary greatly among computers. Most packages still evolve, and weak points present nowadays may be removed tomorrow.

Comparing many packages is a task that can easily be overwhelming. This comparison is based on the analysis of documentation, source programs, and on running examples. Because of space and time limitations, it is hardly comprehensive, and many important issues and features are not discussed. All numbers in the paper that describe limits of various packages or algorithms are approximate. The author apologizes for all omissions and errors that are inevitably a part of all comparisons.

PACKAGES

The main task of packages described here is obtaining solutions to mixed model equations (MME) and estimation of variance components by REML (Henderson, 1984). As a minimum, all programs support animal models with fixed and random cross-classified effects and covariables. Otherwise, the packages differ in many features. In obtaining solutions, the program may be useful for small (< 10,000-100,000 equations) or large (>100,000) problems. The models could be limited to single-trait or could be multitrait. Multitrait models may support missing traits and different models per trait. Other features that may be supported are inbreeding, maternal or unknown parent effects and record weights. As extra functionality, the package may test hypothesis on significance of effects or variance components, or compute approximate prediction error variances for large data sets. In the preprocessing programs, options include ability to recode character fields in the data, validation of animal-parent order and birth dates, elimination of noncontributing animals (pruning) and assignment of unknown parents. The programs may be easy or difficult to install, learn, use or modify. The documentation may be

extensive or minimal. Other characteristics are reliability, sophistication of diagnostics and speed.

Below we briefly characterize the packages covered in this comparison. More information on these packages, as of February 1994, is in the Table and in the later text.

LSML

Dating to 1960s, this program by Harvey (1990) was a forerunner of present evaluation and estimation programs. It computed simple statistics, solutions, variance components and tested hypothesis to mixed models with the diagonal variance-covariance matrices. LSML, which was extensively used and cited until a few years ago, used a dense matrix inversion with absorption (Gaussian-elimination) of one effect, and variance component estimation by Henderson-3 (Henderson, 1984). Although still available, LSML is not discussed later because it supports neither animal model nor REML.

DFREML

Written by Meyer (1988), DFREML was the first public package to implement the derivative-free REML (Smith and Graser, 1986). Extensively cited, it became the standard in the field to which every other program is compared. Its unique feature is a likelihood ratio test for testing the significance of variance component estimates. The documentation is extensive, and it is the only one that has descriptions of all subroutines in the package. DFREML appears very clean of errors. DFREML supports only 10 classes of models, although the important models are included. Also, learning curve is high.

MTDFREML

This program by Boldman, Kriese, Van Vleck and Kachman (1993) has been an independent development and rewrite of an earlier version of DFREML. Compared to current DFREML, MTDFREML is more general model-wise, lacks some features and is easier to install and to use. The MTDFREML manual is user-friendly and contains plenty of program and background information.

PEST & VCE

PEST by Groeneveld, Kovacs and Wang (1990) is an ambitious attempt for a MME solution program with a SAS-like user interface. Its several types of solvers support both small and large data sets. PEST accepts input data in a variety of formats, and is easier to use than other packages. It has been well supported on many computing platforms, is extensively documented and comes with many test files. PEST is the only program that carries an obligatory price and has no comment lines in the program. VCE is a variance component program by Groeneveld (see software paper in this proceedings) that can be used standalone but works best with PEST for data preparation. Many maximization algorithms are supported. VCE is modified and compiled for every model and data by a Unix script.

JAA/MTC etc.

These programs are research off-shots of Misztal. JAA is a solution program that uses iteration-on-data using second-order Jacobi. (Misztal, 1987, 1989). Despite being small, it can process large data sets, and is the only one with support for a large-model approximation of prediction error variances in the animal model. MTC is a REML program using the EM algorithm with canonical transformation and simultaneous diagonalization for support of several random effects (Misztal, 1994). Neither JAA nor MTC supports the maternal effect; MTC does not support missing-traits. JAA was the first animal program used in many countries and has been modified independently to include missing features or to implement a new model. JAA and MTC have fewer features and are less documented than the other packages.

ABTK

The toolkit by Golden, Snelling and Mallinckrodt complements Unix tools and is the only package written in C. Unix tools can perform data manipulation operations such as cutting, pasting, editing, selecting etc., although the number of options in each tool can be intimidating to casual Unix users. Added tools include summation of coefficients, computations of factorization, inverses and traces. Many tools can use compressed files and thus are suitable for large data sets. Tools are combined into programs using pipes and Unix scripts. ABTK installs on Unix only and requires Unix expertise to use it. It also requires knowledge about internals of mixed model computations, although quite good ABTK manuals offers help in many common problems. ABTK is

enthusiastically supported by its developers.

DMU

DMU by Jensen and Madsen is a comprehensive collection of programs used in Denmark for research and routine evaluation. Programs at first appear difficult to use because of cryptic parameter files and insistence on using the same file names in every analysis. However, these nuisances are compensated by extensive diagnostics and completeness of the package. DMU includes a variance component program called DMUAI (based on ideas of R. Thompson) that uses Newton-Raphson maximization. In a single-trait analysis, this program converged in 2 rounds while a DF equivalent took 60. DMU is still being actively modified. When it reaches maturity, it may become the most comprehensive program in this group.

GENERAL CONSIDERATIONS

Models and species. Models used to analyze the data are usually species dependent, and features of packages reflect the species with which the authors work. This orientation can be found by studying examples included with the packages. In dairy cattle, production data are often analyzed by a single trait repeatability model, and conformation data by a multivariate trait (repeatability) model with no missing traits. Unknown parent effects are considered important but inbreeding is not. In studies involving relationships between lactations, a multitrait model with sequentially missing data is desirable. For pig and poultry data, a desirable model is multitrait, and especially for pigs with missing traits. Inbreeding is more important but unknown parents are not. Different model for each trait may be needed. In beef cattle data, models are multitrait with missing data, with maternal effect considered essential. Neither inbreeding nor unknown parent groups are considered important. Required features also depend on the type of data. Experimental data sets with fairly complete pedigrees may benefit more from the consideration of inbreeding but less from unknown parent groups. The opposite is true with large data sets.

Types of programs or packages. Packages are written for many reasons. Some like JAA are research "off-shots" that try to demonstrate a new methodology. These packages have limited functionality and are not necessarily easy to use, but they can do specific tasks (at the time they were written) better than other programs. "Comprehensive" packages like MTDFREML, ABTK or DMU may be written to let a particular person/institution automate common analyses. They tend to be more general, but their ease of use depends on the contact between users and developers, closer contact reducing the need for ease of use. Finally, "universal" packages like PEST try to combine best ideas in the field and be easy to use, comprehensive programs for a general audience. Such packages (may) eventually become commercial. The line between all the categories is blurred, and DFREML can be characterized as the combination of all of them.

A package can also be classified as general-purpose, custom, and toolbox. A general-purpose package supports a class of models. It is not very efficient, but it can be rapidly adapted for various models, within its model range. Most programs fall into this category. A custom package supports just one task, for instance a routine genetic evaluation on a national level. Such a package (e.g., Misztal et al, 1992) is efficient computationally and can provide very elaborate output, but it is difficult to modify. Custom packages are not presented here. The last group, toolboxes such as ABTK, provide a set of "superinstructions". A problem is solved by writing a program in the toolbox language. The toolboxes can be very powerful for many different problems but they require an expertise to write a program.

Ease of use. This criterion can be split into two: ease to learn and ease to use once learned. Only PEST uses a text user interface, where details on models and methods are spelled in quasi-English. DMU uses "cryptic" files that contain a combination of file names and numbers. DFREML, MTDFREML, and JAA ask questions interactively, but their understanding requires prior reading of the manual. Text user interface is definitely easier to use, but it also requires more programming and increases the program size. "Cryptic" parameter files result in a longer learning curve, but do not affect ease of use very much once learned.

Target audience. Application defines what kind of package may be the most desirable. Generality and ease of use are the most important criteria for standard analyses. In research involving new methodology, one would choose programs that are simpler, easier to understand and modify, and where a good support from the author is available.

Larger programs are likely to be more difficult to understand and subsequently modify. If a feature is missing or if a bug is detected, contacting the author may be the only choice. On the other scale, a small program may be missing many features, but adding them could be easy. This makes smaller programs more suitable for "leading-edge" research in new methodology. Alternatively, more comprehensive programs are likely to be preferred by scientists working on standard problems. Good documentation and comment lines greatly simplify modifications of either type of program. All packages except PEST have many comment lines in programs.

TECHNICAL ISSUES

Data recoding. All packages accept data files in at least free format. Most packages PEST can recode all character fields; JAA (via a companion program RENUM) only animals. Other packages accept numeric data only. DFREML and partly JAA prune animal's pedigrees, i.e., eliminate unnecessary animals from pedigrees. JAA verifies that animals are younger than their own parents, and assigns unknown parent groups, both based on the year of birth.

Programming language. All but one package are written in Fortran-77 (F77), which is very efficient in numerical operations, has powerful I/O formatting, and in which extensive numerical libraries are available. Unfortunately F77 has limited syntax and lacks memory management. Consequently large F77 programs are difficult to write, modify, and understand. Also, they need to be recompiled to take advantage of larger memory. Limitations of F77 caused many authors to use common extensions, which work on many but not with all platforms, creating compatibility problems. ABTK uses the C language that addresses many F77 limitations. Not yet released MATVEC by T. Wang (1994, personal communication) uses C++, an object-oriented extension of C. C++ is more difficult to program initially, but with proper libraries combines the ease of use of a matrix package with the versatility and speed of a programming language. There are good public-domain compilers available for C and C++ for almost any computing platform. Because Fortran programs cannot be converted easily to C or C++, a possible upgrade path for existing Fortran-77 programs is via Fortran-90, which supports old Fortran syntax and adds many new features, including memory management and some forms of object programming. However, Fortran-90 compilers are scarce at this time.

Operating system and compilers. Because of extensions incorporated in packages and different implementation of details in compilers etc., programs compiled in one environment may not compile in others. Troublesome extensions include data assignment in the parameter declaration like *integer x/1.0/*, or reading the alphanumeric variable with the * instead of '(a)' format. Many program developers and users have migrated recently to Unix, which has many useful tools, including the *make* tool for program installation. Because Unix systems and compilers are not 100% compatible, distribution prepared on one computer is likely to fail on another one. The author had problems with installation of all packages except JAA, which he developed, and PEST, which is packaged for a specific computer. The package with the most sophisticated script, DFREML, was also the most difficult to install. One should expect the easy installation only on the platform where the package was developed or explicitly tested, however, many problems can be resolved in a few days.

COMPUTING COST

Packages are composed of many blocks such as data preparation, creating MME, computing solutions, and computing the determinant or the inverse of the MME matrix. Usually only one or two blocks are critical to performance. Below we describe some operations and algorithms that are or could be performance bottleneck.

Disk operations. In iterative programs where data or matrix coefficients are read from disk repeatedly, reading from and writing to disk may take up to 95% of all computing time. Therefore, fast disk transfers are essential to high performance. A general rule is that formatted transfers are slower than unformatted, and unformatted transfers are faster for large than small variables. For instance, on a Sparcstation 2, the transfer speed of an integer array of size 3 is 80 kbytes/s formatted and 136 kbytes/s unformatted. For an array of size 100, the transfer speed is 148 formatted and 2007 kbytes/s unformatted. In some compilers the second statement of the functionally identical statements "read(1)(x(i),i=1,100)" and "read(1)x " executes much slower (five times on the

above Sun). All large-data packages use unformatted read, and all of these but PEST use large-variable transfers. *Determinant and traces.* The most expensive procedure in a REML program is the computation of the determinant or trace. DFREML computes the determinant by Gaussian elimination, and the other packages by matrix factorization. JAA and ABTK compute traces by sparse inverse, at a cost of about 3 times larger than the factorization (Misztal and Perez-Enciso, 1983); All reorder the equations by the minimum-degree ordering (George and Liu, 1981). Sparse matrix factorization is faster and more memory-efficient than the sparse matrix absorption because of less overhead, but differences are small. The performance of the sparse computations is very dependent on the sparsity of the matrix, and can be expected proportional to $n^{1.5}ft^2$ memory and $n^2f^2t^3$ arithmetic operations, where n is the number of animals, f is the number of effects, and t is the number of traits. The sparse matrix software allows for matrices as large as 10,000-100,000, depending on the computer, model and data. PEST uses a commercial sparse matrix package, included in its cost. MTDFREML, JAA, ABTK and DMU can use a free package FSPAK (Perez-Enciso et al., 1994). FSPAK has an option to compute the sparse inverse, drastically reducing the cost of implementing the derivative-based REML algorithms.

Derivative and derivative-free maximization in REML. The speed and accuracy of the REML variance component estimation are dependent on the maximization strategy. The popular derivative-free (DF) maximization used in DFREML, MTDFREML, VCE and DMU is very slow in multiple traits as thousands rounds of iteration may be needed to obtain convergence. Assuming that the logarithm of the restricted likelihood function (L) is approximately quadratic, Misztal (1994) calculated that the number of steps to achieve convergence in t -traits is $\sim t^2$ with good DF algorithms, but it does not depend on t in better derivative (D) algorithms. Better DF algorithms included Powell and Rosenbrock and better D algorithms included quasi-Newton or Broyden (as nonlinear solving). The latter include the accelerated EM algorithm. These costs are underestimated because the L function is approximately quadratic only close to the maximum. Worse convergence properties of DF can be seen intuitively by noting that D can sense a desirable direction (gradient) in one round, while DF has to do approximately t^2 rounds to probe all the dimensions. The combined costs of factorization/inversion and maximization are at least t^3 and t^5 numerical operations for better D and DF algorithms, respectively. Programs using D algorithms are not common because DF algorithms are easier to implement and inversion before FSPAK became available was very expensive. The only package that supports multitrait D algorithm (Newton-Raphson) is DMU.

Results of analyses with many traits and the general model are not likely to be accurate. First, the accuracy of the factorization/determinant decreases as traits are becoming more linearly dependent and the MME matrix is larger (Misztal, 94). Second, the maximization method may fail. "Faster" DF (such as Powell or Rosenbrock as opposed to simplex) or D (Newton-Raphson or quasi-Newton as opposed to fixed-point=EM) algorithms may actually converge slower far away from the maximum, and may need a fall-back to slower algorithms in early rounds to avoid divergence. Also, DF's solutions are less accurate than D because finding a maximum, where the maximized function is flat by definition, is less accurate than finding a zero of a derivative, which is not flat. Together, a general-model REML, and particularly DF, may be too expensive and inaccurate

with more than 2-4 traits.

With many traits, the only feasible procedure at this time is canonical transformation (CT) (Lin and Smith, 1990), where the computing cost increases only linearly with the number of traits but only certain models are supported. This procedure is also insensitive to high correlations between the traits, but it is restricted to analyses with same model for each trait, all traits recorded, and one random effect. CT is supported by DFREML, VCE and MTC. Additionally, MTC allows for CT with several random effects by near-simultaneous-diagonalization, extending the range of models to repeatability models (Misztal, 1993). Research by Ducrocq and Besbes (1993) suggests that the other restrictions of CT might be removed.

Figure 1 presents CPU time in general-model D, general-model DF and canonical transformation REML for different t . The times are relatively small for canonical transformation, increase steeply for DF, D being in between. If a single-trait REML took 1 minute of computing time, a 2 trait REML would take at least 1 hr in DF, 8 minutes in D, and 2 minutes in CT. For 5 traits, these times would be 2 days, 2 hrs, and 5 minutes, respectively, and for 15 traits 527 days, 2 days, and 15 minutes, respectively. If the memory required were 2 Mbytes in single trait, for D or DF it would be 8 Mbytes in 2 traits, 50 Mbytes in 5 traits and 450 Mbytes in 15 traits.

Solving systems of equations. Algorithms to solve systems of equations include direct-in-memory, iterative-in-memory, iterative-on-disk, iterative-on-disk-on-data. The direct-in-memory algorithms are those discussed in the section on determinants and traces. They provide accurate solutions but are suitable for small systems of equations (10,000-100,000 equations) because of quadratic cost and substantial memory use. DFREML and MTDFREML use this method exclusively. Iterative methods (Young, 1971) trade speed and accuracy, where a lower accuracy at a lower cost is acceptable; their cost is linear. Memory-based iterative methods are suitable for solving up to 30,000-500,000 equations. PEST implements SOR iteration in memory and DMU can use many different solvers from ITPACK (Kincaid et al., 1982). Solvers in ITPACK calculate acceleration parameters that otherwise have to be provided. Disk-based solvers are slower than those memory-based but are less constrained by the memory limits. ABTK uses a disk-based Gauss-Seidel iteration. It uses the least amount of memory but is consuming a large amount of disk space. In disk-based iterative methods, almost all computing time is spent in disk reading. Because the system of equations is, in general, much larger than the data from which the equations were generated, iteration-on-data, i.e., reading the data and recreating the equation coefficients every round, saves substantial disk space and computing time. The iteration-on-data algorithms can be based on Gauss-Seidel, where several differently sorted data files are needed (Schaeffer and Kennedy, 1986), or on second-order Jacobi where only one

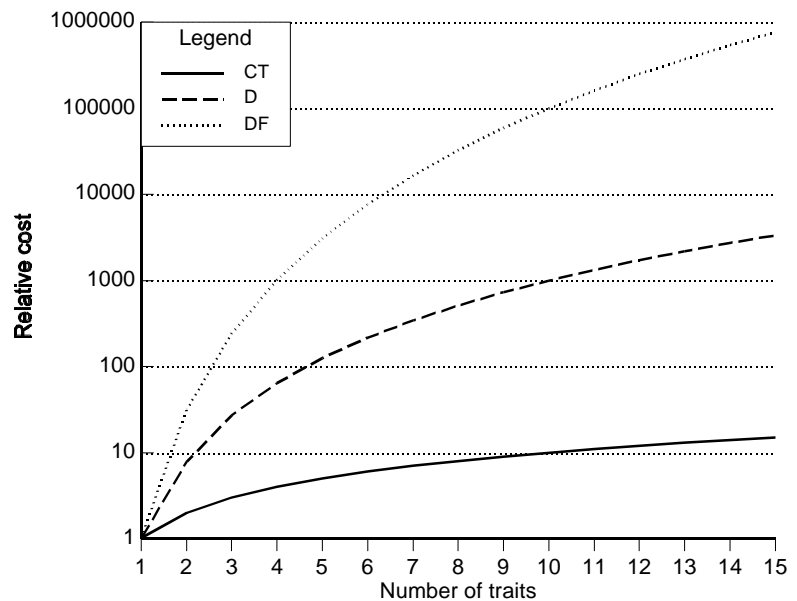


Figure 1. Relative cost of multitrait REML analyses for general-model derivative-free (DF), general-model derivative (D) and canonical transformation (CT) algorithms.

such unsorted file is necessary (Miształ and Gianola, 1987). A combination of the two methods exists where the same file is read several times (Miształ et al., 1989). The iteration-on-data algorithms are suitable for .5 mln equations and more. JAA (single-trait program) and DMU can use iteration on data with a combination of Gauss-Seidel and second-order Jacobi. PEST supports both iteration-on-data solvers.

Multiple traits increase the memory/disk costs of obtaining the solutions iteratively by approximately t to t^2 , and the total computing time may increase even more because of a slower convergence rate. The convergence rate is faster in block iteration, at a cost of more storage for the diagonals (Van Vleck and Dwyer, 1985). The block iteration is supported by PEST and DMU. Storage may be closer to single trait with the implicit storage (Tier and Graser, 1990), and procedure for canonical transformation without commonly known restriction is available (Ducrocq and Besbes, 1993); no package supports either option.

Inbreeding algorithm. The cost of the inbreeding algorithm by Quaas (1976) is quadratic with the number of animals and may be prohibitively expensive with >500,000 animals. Newer algorithms (Tier, 1990; Golden et al., 1991; Meuwissen and Luo, 1992; VanRaden, 1982) have smaller cost. DFREML, PEST, ABTK and DMU support inbreeding; only ABTK uses a faster algorithm.

CONCLUSIONS

This paper compared six packages mainly based on features and algorithms. Another paper in these proceedings will evaluate the packages' accuracy in estimating variance components. We hope that both comparisons will help users to select an appropriate package for their needs. We also hope that comments in this paper will help developers to upgrade their packages.

Most developers are not compensated monetarily for their work. They work hard to ensure that a program written to support a project on a specific platform will work in more general situations under many computing platforms. If we experience problems with installing a public program, we should remember that writing such a program ourselves would take much longer or could be simply impossible. A positive and liberal attitude toward the developers will ensure that new programs/packages will be developed and their authors will not be reluctant to make them widely available.

ACKNOWLEDGEMENTS

Comments by Y. Huang, T. Wang and K.A. Weigel are gratefully appreciated. This paper is possible due to a financial support from the Holstein Association.

REFERENCES

- Boldman, K.G., Kriese, L.A, Van Vleck, L.D. and Kachman S.D. (1993) A manual for use of MTDFREML. USDA-ARS, Clay Center, Nebraska.
- Ducrocq, V. and Besbes, B. (1993) J. Anim. Breed. Genet. 110:81-92.
- George, A. and Liu, J.W.H. (1981) Computer solution of large sparse positive definite systems. Prentice-Hall, Englewood Cliffs, N.J.
- Golden, B.L., Brinks, J.S., Bourdon-R-M. 1991.
- Groeneveld, E., Kovac, M. and Wang, T. (1990) Proc. 4th World Congr. Genet. Appl. Livestock Prod. 13:488-491.
- J. Animal Sci. 69:3564:73
- Harvey, W.R. (1990) User's guide for LSMLMW and MIXMDL. Polycopy.
- Henderson, C.R. (1984) Application of linear models in animal breeding. Guelph, Ontario, University of Guelph.
- Kincaid, D.R., Respes, J.R. Young, D.M. and Grimes, R.G. (1982) ACM Trans. Math. Soft. 8:302.
- Lin, C.Y. and Smith, S.P. (1990) J. Dairy Sci. 73:2494-2502.
- Meuwissen, T.H.E. and Luo, Z. (1992) Genet. Sel. Evol. 24:305-313.
- Meyer, K. (1988) J. Dairy Sci. (Suppl. 2) 71:33-34.
- Miształ, I. and Gianola, D. (1987) J. Dairy Sci. 70:716-723.

- Misztal, I., Gianola, D. and Foulley, J.L. (1989) *J. Dairy Sci.* 72:1557-1568.
Misztal, I., Lawlor, T.J. and Short T.H. (1992) 76:1421-1432.
Misztal, I. (1993) . *J. Dairy Sci.* (Suppl. 1) 76:192.
Misztal, I. and Perez-Enciso, M. (1993) *J. Dairy Sci.* 76:1479.
Misztal, I. (1994) Comparison of computing properties of derivative and derivative-free algorithms in variance component estimation by REML. *J. Animal Breed. Genet.* (submitted).
Perez-Enciso, M., Misztal, I. and Elzo, M.A.. (1994) *FSPAK. Proc. 5th World Congr. Genet. Appl. Livestock Prod.*
Quaas, R.L. (1986) *Biometrics* 32:949-953.
Sedgewick, R. 1984. *Algorithms.* Addison-Wesley, Reading, Mass..
Smith, S. P. and Graser, H.-U. (1986). *J. Dairy Sci.* 69:1156-1165.
Schaeffer, L.R. and Kennedy, B.W. (1986) *Proc. 3rd World Congr. Genet. Appl. Livestock Prod.* 12:382-393.
Tier, N., Graser, H-U. (1990) *J. Animal Breed. Genet.* 108(2):81-88.
Tier, B. (1990) *Genet. Sel. Evol.* 22: 419-430.
VanRaden, P.M. (1982) *J. Dairy Sci.* 75:3136:3144.
Van Vleck, L.D. and D.J. Dwyer. 1985. *J. Dairy Sci.* 68:1006-1014.
Young, D.M. (1971) *Iterative solutions of large linear systems.* Academic Press, New York, NY.

ADDRESSES

- Karin Meyer; Mail: AGBU, University of New England, Armidale, NSW 2351, Australia; E-mail: kmeyer @ didgeridoo.une; Anonymous FTP:metz.une.edu.au
L. D. Van Vleck; Mail: Univ. of Nebraska, Lincoln, NE 68583, USA; E-mail: ansc418@unlvm.bitnet
E. Groeneveld; Mail:FAL, W 3057 Neustadt 1, Hoeltystr.10, Germany; E-mail: eg@jupiter.tzv.fal.d400.de
I. Misztal; Mail: Univ. of Illinois, Urbana, IL 61801, USA; E-mail: ignacy @ uiuc.edu; Anonymous FTP: misz.animal. uiuc.edu.
B.L. Golden; Mail: Colorado State Univ., Fort Collins, CO 80523, USA; E-mail: bgolden@cgel.agsci.colostate.edu; Anonymous FTP: cgel.agsci.colostate.edu.
Just Jensen; Mail: Research Center Foulum, Box 39, 8830 Tjele. Denmark; E-mail: lofjust@vm.uni-c.dk.

Feature ^a	Program/Package					
	DFREML	MTDFREML	PEST /VCE	JAA/MTC	ABTK ^b	DMU
General						
Species orientation	all	beef, pigs	pigs	dairy cattle	beef cattle	all
Year first released	1988	1992	1989/1993	1988/1993	1991	1993
Development system(s)	DEC(Unix)	PC(DOS)	Sun (Unix)	PC (OS/2), Sun (Unix)	RS/6000 (Unix)	Convex(Unix), IBM(CMS)
Lines in source code ^c	21562	9804	12734/4653	7551	>17000	23772
Comment lines in source ^c	2531	2176	6/14	1988	>3000	> 1527
Pages in manual ^c	100	120	102	35	86	24
Functionality						
MME solutions - small data	√	√	√	√	√	√
MME solutions - large data			√	√	√	√
Repeatability - large data				√		
MME - Hypothesis testing	√	√	√			√
REML	√	√	√	√	√	√
SD of REML estimates	√					√
Model						
MT general model	√	√	√		√	√
MT CT	√		√	REML only		
maternal effect	√	√	√		√	√
unknown parent groups	√		√	√		√

^a MME=mixed model equations, MT=multiple trait, CT= canonical transformation; ^b more features supported by programming; ^c approximate