April 14, 2000

Comparison of computing properties of derivative and derivative-free algorithms in variance component estimation by REML

By Ignacy Misztal
University of Illinois, Urbana, Illinois 61801, USA

Introduction

The derivative-free (DF) algorithm (Smith and Graser 1986) is by far the most popular in variance component estimation by REML. Almost all REML software written in the past few years have used this algorithm (Jensen 1993, personal communication; Groeneveld 1993, personal communication; Meyer 1989-91) mainly due to its simplicity. In programming, DF requires only computing the restricted likelihood function along with employment of a maximization algorithm that uses function values alone. In contrast, the derivative algorithms (D), such as EM (Dempster at. al. 1977) or Newton-Raphson (Press et al., 1989) use analytically computed derivatives of the restricted likelihood function. Such derivatives complicate programming, and in certain models are impossible to calculate.

Recent reports suggest that the DF algorithm may be too expensive and/or too inaccurate with more than 1-3 traits. Ducos et al. (1993) used only combinations of 2-trait DF runs in his 7-trait analysis, because 3-trait DF runs were already too expensive. Groeneveld (personal communication, 1994) analyzed a data set containing five-trait records on approximately 9000 animals. The analysis with a general-model program using the DF Simplex algorithm ran for several weeks. In a 5-trait study by Mäntysaari (personal communication) with an 800-animal data set, a DF analysis took close to a day, and the estimates were not plausible. Campos et al. (1993) obtained very different estimates of heritabilities from single and 2-trait analyses with a DF program. Such estimates are expected to be similar.

The first goal of this paper was to investigate whether long running time and numerical inaccuracies in multiple traits are properties of the DF algorithm. The second goal was to compare accuracy and convergence properties of the D and DF algorithms.

Methods

Derivative and derivative-free algorithms.

Let **L** be the restricted likelihood function (Harville 77), and **È** be the vector of variance components. DF denotes a class of maximization algorithms, such as Simplex, Rosenbrock and Powell (Bazaraa et al. 1993; Minoux 1986) that maximize L without using the analytical derivatives:

$$L(\mathbf{\dot{E}}_{\mathbf{max}}) = \max L(\mathbf{\dot{E}}) \qquad \mathbf{(1)}$$

First-derivative algorithms solve (1) using the derivatives computed analytically. For functions that have only one maximum in the search space, these algorithms include those solving the system of nonlinear equations:

$$\mathbf{\dot{E}}_{\mathbf{max}} : \frac{\partial L}{\partial \mathbf{\dot{E}}} = \mathbf{0} \qquad \mathbf{(2)}$$

A popular first-derivative algorithm in REML is EM (Dempster et al. 1977), which can be written as follows:

$$\mathbf{\dot{E}}^{r\ 1} = em(\mathbf{\dot{E}}^{r}) \qquad \mathbf{(3)}$$

where $\mathbf{\dot{E}}^{r}$ presents an estimate of È in the r-th round, and em is a function that returns the next round estimate. The solving algorithm of EM is called fixed point (Woodford 1992), and is considered slow. Faster first-derivative algorithms exist that can be derived from accelerating the EM algorithm (Melijson 1986; Jamshodian and Jennrich 1993). First-derivative algorithms are expected to converge faster than DF because of extra information. Maximization algorithms using second-derivatives, such as Newton-Raphson, are expected to converge even faster, but the complexity of obtaining second-derivatives analytically makes such algorithms less attractive although not impossible. Lately, J. Jensen (personal communication, 1993) implemented a Netwon-Raphson algorithm in REML using R. Thompson's ideas of "average-information."

Convergence of D and DF algorithms

Convergence rate can be described in terms of accuracy as a function of the number of rounds of iteration (Minoux 1986). Linear convergence means linear relationship between accuracy and round of iteration, i.e., constant number of rounds is required for a gain of one extra digit of accuracy. The EM algorithm seems to have a linear convergence. In a superlinear convergence, each round of iteration results in increasingly larger gains in accuracy. Finally, in a n-step convergence, n steps are required to implement one round of iteration.

The convergence rate of maximization algorithms for general functions cannot be derived easily. However, it could be derived for quadratic functions, and any twice-differentiable function, including L, is locally quadratic. Then, the convergence of better DF algorithms such as Powell or Rosenbrock is n-step superlinear (Minoux 1986), and is dependent on n, the dimension of the function being maximized. For better D algorithms, such as quasi-Newton (first-derivative) or Newton-Raphson (second-derivative), the convergence is superlinear and does not depend on n (Bazaraa 1993; Minoux 1986). This leads to conjecture that better DF algorithms converge, in general, n times slower than better D algorithms. N can be expressed in terms of number of random effects, n_r, and number of traits, n_t, as:

$$n = (n_r + 1)n_t(n_t+1)/2 \quad (4)$$

where the extra 1 is for the residual.

This difference in convergence rate is only approximate because differences exist within better D or DF algorithms, and L is not quadratic. Also, more sophisticated methods may fail when the starting point is far away from the maximum.

Cost of one step of D and DF in multiple traits

Let **W** be a coefficient matrix of the mixed model equations Assume that in DF all computing resources are spent in computing $|\mathbf{W}|$. Also assume that in D all computing resources are spent in finding elements of $\mathbf{W}^{-1}$ corresponding to nonzero elements in **W**, or a sparse inverse of **W**. In dense matrices, computing the inverse requires 3 times more arithmetic operations than computing the determinant alone (Duff et al. 1989). Both operations require an equal amount of storage. Approximately the same applies to the number of arithmetic operations in sparse

matrices (Misztal and Perez-Enciso 1993). The amount of storage required is 3 times larger for sparse inversion if the inverse elements are computed by columns (Misztal and Perez-Enciso 1993), but is approximately equal if computing is by rows (R. Thompson, personal communication 1993).

In multiple traits, assuming that almost all traits are recorded, the coefficient matrix of the mixed model equations has n_t times more equations and an average equation has n_t times more nonzero elements than in single traits. In dense matrices, the memory and computing requirements for inversion or computing the determinant for a matrix n_t larger increase as n_t^2 and n_t^3, respectively. The same applies to sparse matrices, where memory and computing requirements increase approximately as pq and pq^2, where p is the number of equations and q is the average number of nonzeroes per equation (Duff et al. 1989).

Cost of one unit of convergence relative to single trait estimation

Let C^1_d be the cost of one step of D in a single trait estimation. Let us compute costs of the same level of convergence in n_t traits for the DF and D algorithms: C^n_{df} and C^n_d. If the DF convergence is n times slower, computations for the matrix operations increase as n_t^3, and computing the determinant costs a third of computing the inverse, the following formulas can be derived:

$$C^n_d = n_t^3 \; C^1_d \tag{5}$$

$$\begin{aligned} C^n_{df} &= \mathbf{a}\ (n_r+1)n_t(n_t+1)/2\ n_t^3\ C^1_d \\ &= (n_r+1)\ n_t^4(n_t+1)C^1_d / 6 \end{aligned} \tag{6}$$

and the relative costs are:

$$C^n_d / C^1_d = n_t^3 \tag{7}$$

$$C^n_{df} / C^1_d = (n_r+1)\ n_t^4(n_t+1)/6 \tag{8}$$

$$\approx (n_r+1)\ n_t^5/6 \tag{9}$$

$$C^n_{df} / C^n_d = (n_r+1)\ n_t(n_t+1)/6 \tag{10}$$

$$\approx (n_r + 1)\, n_t^2/6 \qquad (11)$$

According to equation (7), the number of numerical operations in D increase cubically with the number of traits. In equation (8), the cost of DF increases with the fifth power of the number of traits. From equation (1) one can find that the costs of DF and D in models with 2 random effects are similar in single trait. DF is less expensive with 1 random effect, and D is less expensive with more than 2 random effects. In multiple traits, DF is n_t^2 more expensive than D.

Relative costs of multitrait DF REML evaluation using DF and D algorithms, computed with formulae (7) and (8) are presented in Table 1. The number of numerical operations increases rapidly in multiple traits, and for 2, 3, 4 and 5 traits, DF is 24, 162, 640 and 1875 times more expensive than a single trait D, respectively. For D the corresponding increases are smaller than DF but still large overall: 8, 27, 64 and 125, respectively. Although the memory increases are smaller: 4 , 9, 16 and 25, respectively, they may be prohibitively large considering that often insufficient memory limits the size of datasets in REML analyses.

The steep increases in the computing requirements in DF in multiple traits explain high running times of DF programs mentioned in the introduction. If a single-trait DF analysis took one minute of computer time, a 2, 3, 4 and 5-trait DF analysis would take approximately .5, 2.5, 11 and 31 hours, respectively. A run of D would take .1, .5, 1, and 2 hours, respectively. If a single-trait run took 1 hour, a 2, 3, 4 and 5-trait DF analysis would take approximately 1, 7, 27 and 78 days, respectively, and a run of D would take .3, 1, 3 and 5 days, respectively.

Accuracy of the D and DF algorithms

In DF, if L(**È**) is computed with r significant digits, then **È** can be bracketed with at most r/2 significant digits (Press et al. 1989). Such a limit does not exist in D. Worse numerical accuracy of the DF maximization is illustrated with the help of Figure 1, which shows a quadratic function and its derivative. The function is very flat, and the maximum of the function cannot be determined accurately by looking at the function alone. The maximum can be determined much more precisely by finding a zero of the function's first derivative.

Loss of accuracy in DF does not appear a problem at first. Most computations are done

with double precision, which corresponds to 15-17 significant decimal digits, and estimates with 1% or 2 significant digits of accuracy are considered sufficently accurate in practice. However, the likelihood function could have low accuracy for a variety of reasons. Some accuracy is lost due to a roundoff error when summing all components of L. Components of that function may have reduced accuracy. This particularly applies to the determinant, where the loss of accuracy could result from lack of pivoting in Cholesky decomposition-based packages, poor conditioning of the coefficient matrix and rounding errors associated with computing in large matrices. In multiple traits, the coefficient matrix of the mixed model equations is composed of $R_0^{-1}$and G_{i0}^{-1} - covariance matrices between traits for residual and random effect i, respectively. Poor conditioning of these matrices results in poor conditioning of W and subsequently low accuracy of determinants and traces. $R_0^{-1}$and G_{i0}^{-1} would be poorly conditioned numerically when traits are highly correlated or linearly dependent. The loss of accuracy due to poor conditioning of **W** is also present in derivative maximization, but it has smaller effect on the estimates because of better numerical properties of the derivative maximization.

Another source of inaccuracy could arise in algorithms where derivatives are obtained numerically by differentiation. For example, the Quasi-Netwon algorithm can be so implemented in DF, and one can regard other DF algorithms as using the numerical differentiation implicitly .The accuracy of such differentiation is dependent on the step side and could be very low for steps too large or too small. Subsequently, the accuracy would be dependent on parameters that define the step size, and in particular could be good for some problems but poor for others.

Canonical transfomation

In general-model REML, such as discussed previously, the computing costs for one round of iteration or function evaluation increase cubicly with the number of traits. In canonical transformation (CT) (Thompson 1976), such an increase is only linear, and memory costs increase very little. D seems to be better suited than DF as a single-trait algorithm within CT because each round of iteration in CT could be regarded as a restart and restarts in DF are expensive. With D in CT, the convergence rate should not be much different from general-model D. Consequently the cost of CT could be linear with respect to the number of traits. Numerical

properties of CT are better than in general models because of implicit scaling and easy taking care of a nearly-singular (co)variance matrices in case of high correlation between traits. CT was believed to be applicable only to a narrow class of models but Ducrocq and Besbes (1993) have shown that almost all restrictions of the canonical transformation while estimating breeding values could be removed. Misztal (1993) extended CT REML to multiple-random effects. Further research will determine if other CT restrictions can be removed in REML.

Numerical tests

Data consisted of 4540 records on 2147 cows with 3 conformation scores. Model included 51 fixed management, 2147 random permanent environmental and 2697 animal effects. The choice of data set was not expected to influence the results significantly. Variance components for 1 to 3 traits were computed with several programs. The first set of programs were DFUNI for single-trait and DFMUV for multiple-traits from package DFREML version 2.1 by Karin Meyer (Meyer 1991; Misztal 1994). Both programs implemented DF Powell (later called DF/Powell) and DF Simplex (DF/Simplex) algorithms. Runs with the Simplex algorithm used default parameters and runs with the Powell algorithm had stopping criteria decreased from deafult 10^{-4} to 10^{-6}. Because of numerical problems with more complex options, multi-trait analyses used an option for equal design matrices. The second program was DMUEM that used an accelerated EM algorithm (later called AEM). This program was E. Mäntysaari's modification of DMUAI of package DMU (Misztal 1994) originally written by Jensen and Madsen with support for the Newton-Raphson maximization. Acceleration in DMUEM was by Aitken algorithm applied every several rounds when acceleration parameters became sufficiently stable. DMUAI was not evaluated here because its debugging has not been finished when this paper was written. The last program was MTC (later called CT) by Misztal (Misztal 94) that used a canonical transformation, the EM algorithm, and multiple diagonalization for support of multiple random effects. With multiple diagonalization the estimates are only approximate, but the accuracy of the approximation was found very high for conformation and type traits (Misztal 1993). MTC was not a general-model program because it required equal design matrices and all traits recorded. Stopping criteria for MTC and DMUEM were relative-averaged quadratic

changes between subsequent rounds smaller than 10^{-8}. Starting variances for DMUEM were 60% of estimates computed by MTC, with covariances set to 0. For DFREML such covariances were set to a number close but not equal to 0. DFMUV does not estimates covariances if their prior values are 0. MTC was started with variance ratios 1 for all traits.

Results and Discussion

When priors for covariances were close to 0, DF/Simplex diverged in 2 traits while DF/Powell required large number of likelihood evaluations. To avoid execessively high running time and divergence, these priors were set so that all initial correlations between traits were 50%. In all analyses, including those unreported with different priors, AEM and EM always converged predictably to several decimal digits of accuracy, and convergence rate was much less dependent on the choice of priors.

Table 2 shows the number of likelihood evaluations or rounds of iteration, computer time, estimates of heritabilities and genetic correlations for 1 to 3-trait analyses. Regarding the DMUEM estimates as exact REML, all computing options provided accurate estimates in single trait, differed at most by.01 (DF/Simplex by.02) for 2 traits, and were up to .07 off for both DF options in 3 traits. MTC's estimates were off by at most .01 in all traits except for one .03 heritability difference in 3 traits. Errors in MTC were due to approximation of multiple diagonalization and not to the lack of convergence. The accuracy of the DF programs might have been better if their algorithms had been restarted or parameters of their maximization algorithms had been tuned. Experiences with such a tuning are discussed by Boldman et al. (1993) and Kovac (1991).

Relative numbers of likelihood evaluations for DF and rounds of iteration for AEM/CT are given in Table 3. For CT, the convergence rate was practically independent of the number of traits. For AEM, the convergence rate was 1.4 times slower for 2 traits and 1.8 times slower for 3 traits. For DF, the convergence rate was slower in 2 and 3 traits by 9 and 20 times for Powell, respectively, and by 13 and 57 times for Simplex, respectively. The dependence of convergence rate on the number of traits for DF was closer to 3-rd or 4-th power than expected quadratic,

being larger for Simplex than for Powell. Larger increases for Simplex agree with Bazaraa et al. (1993), where this algorithm was described as less efficient when the dimensionality of the optimization increased. Only the convergence of the Powell algorithm was expected to be n-step superlinear. Slower than expected convergence in DF and some decrease in convergence rate of AEM could be caused by deviation of L from the quadratic function.

Table 4 shows increase of computing time per one likelihood evaluation or round of iteration. Rather than expectic cubic, the dependence was between linear and quadratic for DF and between quadratic and cubic for AEM. Such an increase in CT was less than linear. Smaller than expected increase in computing time for DF and AEM can be explained by two mechanisms. First, DFMUV used as DF took advantage of equal design matrices for each trait and subsequently the mixed-model matrix was more sparse. Such simplification would not have been possible if data contained missing traits or unequal models were used per trait. In such a case, DF would be about 5 times more expensive in 2 traits and 15 times in 3 traits. Second, operations other than sparse matrix factorization or inversion, such as setting-up the coefficient matrix, have costs lower than cubic. They are not negligible for small problems.

Computing time relative to the single trait DF/Powell is shown in Table 5. In single traits, DF/Powell was the least expensive program, with DF/Powell, AEM and CT running 1.8, 4.2 and 9.6 times longer, respectively. In 2 traits, CT was the least expensive one at a relative cost of 17 , followed by AEM at 22, DF/Powell at 26, and DF/Simplex at 71. In 3 trait runs, CT was the least expensive at 21, followed by AEM at 88, DF/Powell at 151, and DF/Simplex at 686. AEM was less expensive in multiple traits than any of the DF algorithms. The numbers from Table 5 should be treated cautiously because of incompatible features of the programs. Single-trait DF runs used a single-trait program DFUNI, which did not have any multiple-trait overhead present in CT and AEM. CT time was particularly slow in single trait because it used an unaccelerated EM algorithm. With acceleration, the CT convergence could be similar to that of AEM and computing times for all traits would be reduced accordingly. For instance, a single-trait accelerated version of CT converged in 25 rounds in 75 s, and ran only 1.8 longer than DF/Powell or as fast as DF/Simplex. Finally, despite a similar convergence criterion, the level of accuracy obtained by DF was smaller than by the other programs. For similar level of real accuracy (as

measured by differences from exact estimates) CT and AEM could have been iterated 30-50% fewer rounds.

Times relative to those of single-trait runs are shown in Table 6. The increase was less than linear for CT, between quadratic and cubic for AEM, almost fifth-power for DF/Powell, and even larger for DF/Simplex. The increases for DF/Powell and AEM are similar to those expected from better DF and D algorithms in Table 1.

The comparison serves as a review of issues rather than a guideline to selecting a particular program. Only one data file and one model was used, and only results for 1 or 2 priors are reported. Only a few options in DFREML were tried and in particular the runs were not restarted. The programs differ greatly by the level of finishing. DFUNI and DFMUV were very well finished, culuminating years of interest and research in DF REML algorithms. New interest in D REML programs was generated only recently after sparse-matrix inversion software became available, and DMUEM or MTC are still under active development.

Conclusions

Convergence of DF algorithms is strongly dependent on the number of traits, type of algorithm, algorithm details, and priors. Subsequently, in mutiple traits these algorithms are not only expensive but also unreliable, and reliable estimates should not be expected from more than 2-4 traits. The convergence of D algorithms almost does not depend on the number of traits; reliable convergence was achieved when DF algoritms failed. Well programmed D algorithms have potential to be be faster than DF algorithms in almost all cases. Despite properties better than DF, D algorithms are prohibitively expensive with many traits. For instance, in a 5-trait analysis, the D algorithm will require about 125 more computing time and 25 times more memory than in a single trait. With a large number of traits, the only feasible procedure at this time is canonical transformation, where computing costs increase approximately linearly with the number of traits but only certain models are supported. Further research will determine whether the canonical transformation REML can be generalized to general models.

Summary

Computing properties of better derivative and derivative-free algorithms were compared theoretically and practically. Assuming that the log-likelihood function is approximately quadratic, in a t-trait analysis the number of steps to achieve convergence increases as t^2 in "better" derivative-free algorithms and is independent of that number in "better" derivative algorithms. Cost of one step increases as t^3. Subsequently, both classes of algorithms have similar computational cost for single trait models. In multiple traits the computing costs increase as t^3 and t^5, respectively. The derivative-free algorithm is worse numerically conditioned. Four programs were used to obtain 1, 2 and 3 trait REML estimates from field data. Compared to single trait analyses, the cost of one run for derivative-free algorithms increased by 27-40 times for 2 traits and 152-686 for 3 traits. Similar increase in rounds of iteration for a derivative algorithm was 5 and 21, and it was 1.8 and 2.2 in canonical transformation. Convergence and estimates of derivative algorithms were more predictable, and unlike derivative-free algorithms, were not dependent on the choice of priors. Well implemented derivative REML algorithms are less expensive and more reliable in multiple traits than derivative-free ones.

Acknowledgements

This study was supported by a grant from the Holstein Association of America. Suggestions by R. Fernando, K. Marshall, R. Tempelman, T. Wang and the anonymous referee as well as programs by J. Jensen, E. Mäntysaari and K. Meyer are greatfully acknowledged

References

Bazaraa, M.S.; Sherali H. D.; Shetty C.M., 1993: Nonlinear programming. John Wiley & Sons, New York.

Boldman, K.G.; Kriese, L. A.; Van Vleck, L. D.; Kachman, S. D., 1993: A manual for use of MTDFREML. USDA-ARS, Clay Center, Nebraska.

Campos, M.S.,Wilcox, C.J., Becerril, C.M., Diz, A. 1994. Genetic parameters for yield and reproductive traits of Holstein and Jersey cattle in Florida. J. Dairy Sci. 77:867-873.

Dempster, A. P.; N. M. Laird; and D. B Rubin, 1977: Maximum likelihood from

incomplete data via the EM algorithm. Proc. R. Stat. Soc. B 39:1.

Ducos, A.; Bidanel, J.P.; Ducrocq, V.; Boichard ,D.; Groeneveld, E., 1993: Multivariate restricted maximum likelihood estimation of genetic parameters for growth, carcass and meat quality traits in French Large White and French Landrace pigs. Genet. Sel.

Evolut. 25:475-493 .

Ducrocq, V.; Besbes, B., 1993: Solution of multiple trait animal models with missing data on some traits. J. Anim. Breed. Genet. 110:81-92.

Duff, I.S.; Erisman, A.M.; Reid, J.K, 1989: Direct methods for sparse matrices. Clarendon Press. Oxford.

Harville, D.A., 1977: Bayesian interference for variance component estimation and to related problems. J. Am. Stat. Assoc. 72:320.

Jamshidian, M.; Jennrich, R.I., 1993: Conjugate gradient acceleration of the EM algorithm. J. American Stat. Associ. Vol. 88 421:221-228.

Kovac, M., 1991. Derivative free methods in covariance component estimation. Ph.D. Diss., Univ. Illinois, Urbana, USA.

Laird, N. M.; Lange, N.; Stram D., 1987: Maximum likelihood computations with repeated measures: application of the EM algorithm. J. American Stat. Associ. 82:97.

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., 1990: Present status of knowledge about statistical procedures and algorithms available to estimate variance and covariance components. In: Proceedings of the 4th World Congress Applied to Livestock Production, Edinburgh, Scotland. XIII:407:418.

Meyer, K., 1991: Estimating variances and covariances for multivariate animal models by restricted maximum likelihood. Genet. Sel.

Evol. 23:67-83.

Meillijson, I., 1989: A fast improvement to the EM algorithm on its own terms. J. R. Statist. Soc. B 51. No. 1:127-138.

Misztal, I., 1993: Multitrait REML algorithm in repeatability models. J. Dairy Sci. (Suppl. 1) 76:192.

Misztal, I., 1994: Software packages in animal breeding. Proc. 5th World Cong. Genet. Livest. Prod., Guelph (in print).

Misztal, I; Perez-Enciso, M., 1993: Sparse matrix inversion for restricted maximum likelihood estimation by expectation-maximization. J. Dairy Sci. 76:1479-1483.

Minoux, M,. 1986: Mathematical programming. John Wiley & Sons, Chichester.

Perez-Enciso, M.; Misztal I., 1993: FSPAK - An interface for public domain sparse matrix subroutines. Manuscript. (anonymous FTP at 128.174.78.6).

Press, W. H.; Flannery, B.P.;Teukolsky, S.S.; Vetterling, W.T., 1989: Numerical recipes. Cambridge University Press, Cambridge.

Smith, S. P.; Graser, H.-U., 1986: Estimating variance components in a class of mixed models by restricted maximum likelihood. J.

Dairy Sci. 69:1165.

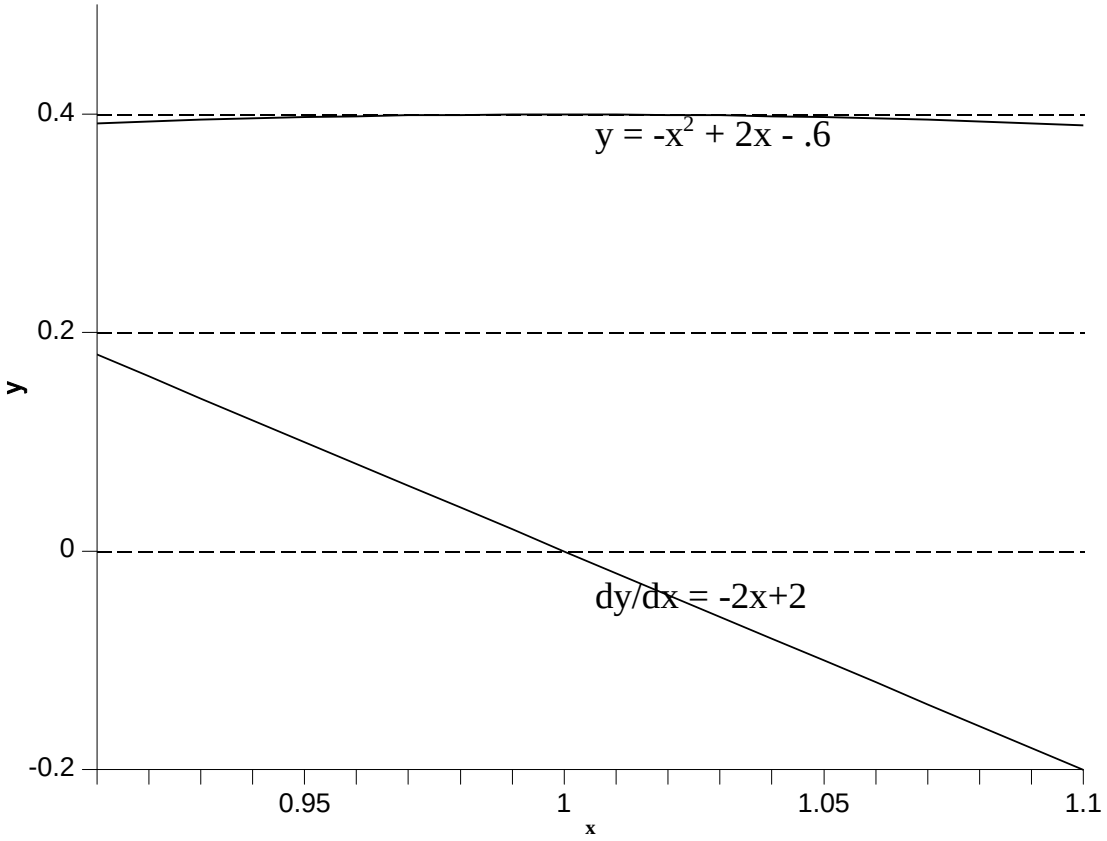
Thompson, R., 1976: Estimation of quantitative genetic parameters. In Proc. Intern. Conf.Quantitative Genetics. E. Pollak, O. Kempthorne, and T. B. Bailey (eds.), Ames, Iowa, 639:657.

Woodford, C., 1992: Solving linear and non-linear equations. Ellis Horwood, Chichester.

Author's adress: *I. Misztal, University of Illinois, 1207 W. Gregory Dr., Urbana, IL 61801, USA.*

Author's electronic mail address: ignacy@uiuc.edu

Figure 1. A quadratic function and its derivative[1].



[1] The maximum is found by looking at y alone in derivative-free maximization and by finding zero of dy/dx in first-derivative maximization.

Table 1. Theoretical relative number of arithmetic operations and memory requirements of derivative free (DF) and Derivative (D) algorithms for a 2 random-effect model.

Number of traits	Number of arithmetic operations		Memory requirements
	DF	D	
1	1	1	1
2	24	8	4
3	162	27	9
4	640	64	16
5	1875	125	25
6	4536	216	36

Table 2. Number of rounds (likelihood evaluations for DF), computing time, estimates of heritabilities and genetic correlations in 1 to 3 trait analyses for programs with derivative free Powell optimization (DF/Powell), derivative-free Simplex optimization (DF/Simplex), accelerated EM, and extended canonical transformation (CT).

Number of traits	Measure	Values			
		DF/Powell	DF/Simplex	AEM	CT
1	rounds	26	47	24	85
	time [s]	42	76	175	405
	h_1^2	.45	.45	45	.45
2	rounds	238[ab]	639[ac]	33	84
	time [s]	1129	2997	920	672
	h_1^2	.44	.47	.45	.44
	h_2^2	.33	.35	.33	.34
	$r_{g1,2}$	.76	.80	.76	.77
3	rounds	583[a]	2696[ad]	45	81
	time [s]	6380	28808	3679	898
	h_1^2	.40	.45	.44	.44
	h_2^2	.26	.26	.32	.35
	h_3^2	.40	.37	.41	.41
	$r_{g1,2}$	.70	.72	.76	.76
	$r_{g1,3}$	.74	.72	.78	.79
	$r_{g2,3}$	.99	.93	.95	.95

[a] With prior correlations between traits for all effects .50.

[b] Needed 2024 likelihood evaluations when prior correlations bewteen traits $\approx$ 0

[c] Diverged after 1700 likelihood evaluations when prior correlations bewteen traits $\approx$ 0

d Partial convergence with simplex variance at $.4 \times 10^{-4}$

Table 3. Relative number of likelihood evaluations or rounds of iteration for 4 computing options and 1-3 traits.

Number of traits	Relative number of likelihood evaluations or rounds of iteration			
	DF/ Powell	DF / Simplex	AEM	CT
1	1.0	1.0	1.0	1.0
2	9.2	13.6	1.4	1.0
3	20.7	57.4	1.8	1.0

Table 4. Relative computing time per likelihood evaluations or round of iteration for 4 computing options and 1-3 traits.

Number of traits	Relative time per likelihood evaluation or round of iteration [s]			
	DF/ Powell	DF / Simplex	AEM	CT
1	1.0	1.0	1.0	1.0
2	2.5	2.5	3.8	1.7
3	6.7	6.7	11.2	2.3

Table 5. Computing time relative to single-trait DF/Powell for 4 computing options and 1-3 traits .

Number of traits	Relative Computing Time			
	DF/ Powell	DF / Simplex	AEM	CT
1	1.0	1.8	4.2	9.6
2	26.8	71.3	21.9	16.8
3	151.9	685.9	87.6	21.3

Table 6. Relative computing time for 4 computing options and 1-3 traits.

Number of traits	Relative Computing Time			
	DF/ Powell	DF / Simplex	AEM	CT
1	1.0	1.0	1.0	1.0
2	26.8	39.6	5.3	1.8
3	151.9	381.0	21.0	2.2