Computing procedures for genetic evaluation including phenotypic, full pedigree, and genomic information

I. Misztal, Andres Legarra, Ignacio Aguilar

J. Dairy Sci. 92:4648–4655
doi:10.3168/jds.2009-2064

Computing procedures for genetic evaluation including phenotypic, full pedigree, and genomic information

I. Misztal,*[1] A. Legarra,† and I. Aguilar*‡
*Department of Animal and Dairy Science, University of Georgia, Athens 30602
†Institut National de la Recherche Agronomique (INRA), UR631 SAGA, BP 52627, 32326 Castanet-Tolosan, France
‡Instituto Nacional de Investigación Agropecuaria, Las Brujas 90200, Uruguay

ABSTRACT

Currently, genomic evaluations use multiple-step procedures, which are prone to biases and errors. A single-step procedure may be applicable when genomic predictions can be obtained by modifying the numerator relationship matrix $\mathbf{A}$ to $\mathbf{H} = \mathbf{A} + \mathbf{A}_\Delta$, where $\mathbf{A}_\Delta$ includes deviations from expected relationships. However, the traditional mixed model equations require $\mathbf{H}^{-1}$, which is usually difficult to obtain for large pedigrees. The computations with $\mathbf{H}$ are feasible when the mixed model equations are expressed in an alternate form that also applies for singular $\mathbf{H}$ and when those equations are solved by the conjugate gradient techniques. Then the only computations involving $\mathbf{H}$ are in the form of $\mathbf{Aq}$ or $\mathbf{A}_\Delta\mathbf{q}$, where $\mathbf{q}$ is a vector. The alternative equations have a nonsymmetric left-hand side. Computing $\mathbf{A}_\Delta\mathbf{q}$ is inexpensive when the number of nonzeros in $\mathbf{A}_\Delta$ is small, and the product $\mathbf{Aq}$ can be calculated efficiently in linear time using an indirect algorithm. Generalizations to more complicated models are proposed. The data included 10.2 million final scores on 6.2 million Holsteins and were analyzed by a repeatability model. Comparisons involved the regular and the alternative equations. The model for the second case included simulated $\mathbf{A}_\Delta$. Solutions were obtained by the preconditioned conjugate gradient algorithm, which works only with symmetric matrices, and by the bi-conjugate gradient stabilized algorithm, which also works with nonsymmetric matrices. The convergence rate associated with the nonsymmetric solvers was slightly better than that with the symmetric solver for the original equations, although the time per round was twice as much for the nonsymmetric solvers. The convergence rate associated with the alternative equations ranged from 2 times lower without $\mathbf{A}_\Delta$ to 3 times lower for the largest simulated $\mathbf{A}_\Delta$. When the information attributable to genomics can be expressed as modifications to the numerator relationship matrix, the proposed methodology may allow the upgrading of an existing evaluation to incorporate the genomic information.

Key words: best linear unbiased predictor, genomic selection, single nucleotide polymorphism, genetic evaluation

INTRODUCTION

Availability of dense molecular markers of type SNP led to the recent introduction of the genome-wide or genomic selection evaluation models. Those models are most often based on the simultaneous estimation of SNP marker effects $\mathbf{a}$. Differences among methods are mostly on the a priori distribution of marker effects $\mathbf{a}$ (Meuwissen et al., 2001; Gianola et al., 2006). Efficient procedures exist for the computation of $\mathbf{a}$, even for large data sets (Legarra and Misztal, 2008).

The genomic evaluation is currently implemented as a multistep procedure. For example, an implementation for US dairy cattle (VanRaden, 2008; VanRaden et al., 2009) requires 3 steps: a) regular evaluation by the animal model, b) estimation of genomic effects for a relatively small number of genotyped animals, and c) estimation of genomic breeding values by a selection index. The elements in the index include a parent average or PTA from step a), genomic solutions from step b), and a parent average or PTA computed based on genotyped ancestors. Weights in the index are functions of heritability and accuracy. The marker-assisted selection program in France simultaneously fits QTL and polygenic effects, with weights depending on associated variance components (Guillaume et al., 2008).

Advantages of the multistage procedure include no change to the regular evaluations and simple steps for predicting genomic values for young genotyped animals. Disadvantages are requirements for parameters in steps b) and c) such as prior variances and weights, and loss of accuracy and biases attributable to selection. Whereas the model in a) uses the information on all animals and can be multitrait, the model in b) is equivalent to a single-trait sire model for a highly selected set of sires. Incorrect parameters in b) and

Received January 26, 2009.
Accepted April 29, 2009.
[1]Corresponding author: ignacy@uga.edu

c) can result in unexpected changes for high-reliability bulls. Neuner et al. (2008) claimed that problems associated with the multistep procedure reduce its benefits, especially for cows.

VanRaden (2008) investigated 2 options for step b): "nonlinear," based on estimating effects attributable to SNP markers with a prior mixture distribution for those effects, and "linear," based on prior normal distribution for SNP markers. The latter is equivalent to using mixed model equations with a genomic relationship matrix. For most dairy traits, predictions based on the estimation of marker effects with nonlinear predictions were practically equivalent to linear predictions and thus to predictions with BLUP using a genomic relationship matrix (Cole et al., 2009; VanRaden et al., 2009). Therefore, using the genomic relationship matrix results in little or no loss of accuracy.

One way to simplify the multistep procedure is by incorporating the genomic information into step a), resulting in a single-step procedure. This could be accomplished by modifying the numerator relationship matrix $\mathbf{A}$ in that evaluation to include the genomic information. Such modifications are presented and discussed by Legarra et al. (2009) in a companion paper.

Assume that such a modification is known and that it involves relatively few elements of $\mathbf{A}$. The mixed model equations require $\mathbf{A}^{-1}$, which is very easy to create for large populations because of its sparsity and its special structure (Henderson, 1976). However, obtaining the inverse of the modified matrix is likely to be impossible in general for large populations. This is not only because the cost of inversion is high, but also because $\mathbf{A}$ is dense and thus too large to store for large pedigrees. Thus, an approach using a modified $\mathbf{A}$ is of little value unless a feasible computing approach is available. The purpose of this study is to develop an efficient computing strategy to obtain solutions to mixed model equations in which the numerator relationship matrix is modified by a known matrix accounting for the genomic information.

MATERIALS AND METHODS

Assume regular mixed model equations as used in a traditional genetic evaluation, for simplicity with only a single random effect:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{Zu} + \mathbf{e},$$

where $\mathbf{y}$ is a vector of records, $\mathbf{b}$ is a vector of fixed effects, and $\mathbf{u}$ is a vector of animal effects. Under a polygenic infinitesimal model of inheritance, $\text{var}(\mathbf{u}) = \mathbf{A}\sigma_\alpha^2$, where $\mathbf{A}$ is the numerator relationship matrix based on pedigree. Furthermore, $\text{var}(\mathbf{e}) = \mathbf{I}\sigma_e^2$, and $\mathbf{X}$ and $\mathbf{Z}$ are appropriate incidence matrices.

Assume that the numerator relationship can be modified to account for genomic information:

$$\mathbf{H} = \mathbf{A} + \mathbf{A}_\Delta,$$

where $\mathbf{A}_\Delta$ is a matrix that can be stored explicitly, and $\mathbf{H}$ is the new modified matrix. In the simplest case, a genomic relationship matrix $\mathbf{G}$ replaces the numerator relationship matrix for the genotyped animals. Let indices 1 and 2 refer to ungenotyped and genotyped animals, respectively. Then

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix},$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{G} \end{bmatrix} = \mathbf{A} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} - \mathbf{A}_{22} \end{bmatrix},$$

and

$$\mathbf{A}_\Delta = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} - \mathbf{A}_{22} \end{bmatrix}.$$

Legarra et al. (2009) proposed several $\mathbf{H}$ based on the partition of animals into several groups, including ungenotyped and genotyped animals. Although their different $\mathbf{H}$ are more complex than in the simple case, most quantities can be computed efficiently without any steps involving large matrix multiplications. Therefore, for simplicity of presentations, the following computing formulas assume the simple case above.

Solving Algorithm

Assume that $\mathbf{G}$ and $\mathbf{A}_{22}$ are available. Temporarily assume that $\mathbf{H}$ is positive definite. The regular mixed model equations are

$$\begin{bmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{Z} \\ \mathbf{Z}'\mathbf{X} & \mathbf{Z}'\mathbf{Z} + \alpha\mathbf{H}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{y} \\ \mathbf{Z}'\mathbf{y} \end{bmatrix}$$

or

$$\mathbf{LHS}\ \mathbf{w} = \mathbf{RHS}$$

using the usual notation, where $\mathbf{LHS}$ and $\mathbf{RHS}$ are the left- and right-hand side, and $\mathbf{w} = \begin{bmatrix} \hat{\mathbf{b}}' & \hat{\mathbf{u}}' \end{bmatrix}'$.

Assume that the system of equations is solved using an algorithm that does not require the elements of **LHS** explicitly but only its product by a vector, say **LHS q**, as in the preconditioned conjugate gradient (**PCG**) iteration on data (Tsuruta et al., 2001). Then

$$\mathbf{LHS\ q} = \begin{bmatrix} \mathbf{X'Xq_1 + X'Zq_2} \\ \mathbf{Z'Xq_1 + Z'Zq_2 + \alpha H^{-1} q_2} \end{bmatrix} = \begin{bmatrix} \mathbf{c_1} \\ \mathbf{c_2 + c_3} \end{bmatrix},$$

with

$$\mathbf{q} = \begin{bmatrix} \mathbf{q_1} \\ \mathbf{q_2} \end{bmatrix};\ \mathbf{c_2} = \mathbf{Z'Xq_1 + Z'Zq_2};\ \mathbf{c_3} = \alpha \mathbf{H^{-1} q_2};\ \mathbf{RHS} = \begin{bmatrix} \mathbf{r_1} \\ \mathbf{r_2} \end{bmatrix}.$$

However, $\mathbf{H}^{-1}$ can be computed only for small populations; furthermore, **H** might be singular or close to singularity. Henderson (1984, 1985) and Harville (1976) described an unsymmetric set of mixed model equations in which only **H**, not necessarily of full rank, is required:

$$\begin{bmatrix} \mathbf{X'X} & \mathbf{X'Z} \\ \mathbf{HZ'X} & \mathbf{HZ'Z} + \alpha \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X'y} \\ \mathbf{HZ'y} \end{bmatrix}$$

or

$$\mathbf{LHS_M w} = \mathbf{RHS_M}.$$

For that set,

$$\mathbf{LHS_M\ q} = \begin{bmatrix} \mathbf{X'Xq_1 + X'Zq_2} \\ \mathbf{HZ'Xq_1 + HZ'Zq_2 + \alpha q_2} \end{bmatrix} = \begin{bmatrix} \mathbf{c_1} \\ \mathbf{Hc_2 + \alpha q_2} \end{bmatrix} = \begin{bmatrix} \mathbf{c_1} \\ \mathbf{Ac_2 + A_\Delta c_2 + \alpha q_2} \end{bmatrix}$$

with

$$\mathbf{RHS_M} = \begin{bmatrix} \mathbf{r_1} \\ \mathbf{Hr_2} \end{bmatrix} = \begin{bmatrix} \mathbf{r_1} \\ \mathbf{Ar_2 + A_\Delta r_2} \end{bmatrix},$$

The new formulas do not include $\mathbf{H}^{-1}$ but include $\mathbf{A_\Delta c}_2$, $\mathbf{Ac}_2$, $\mathbf{A_\Delta r}_2$, and $\mathbf{Ar}_2$. For the simplistic **H**, the first term can be computed directly at a low cost. The second term can also be computed inexpensively following the algorithm by Colleau (2002; see Appendix A), which uses only the pedigree information and is completed in the amount of time proportional to the number of animals. The same algorithm also can be used to compute $\mathbf{Ar}_2$. Selected elements of **A** can be computed recursively, for example, by using the algorithm by Aguilar and Misztal (2008).

More Complicated Models

Assume a multiple trait model, possibly with effects such as random regression or maternal. The regular mixed model equations for such models can be presented as

$$\begin{bmatrix} \dots & \dots \\ \dots & \dots + \mathbf{G_{0a}^{-1}} \otimes \mathbf{H^{-1}} \end{bmatrix} \begin{bmatrix} \dots \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \dots \\ \mathbf{Z'R^{-1}y} \end{bmatrix},$$

where parts not listed (...) are due to effects other than $\hat{\mathbf{u}}$. By expanding the unsymmetric model by Henderson (1984) to multiple traits, the quantities needed for the iterations become

$$\mathbf{LHS_M q} = \begin{bmatrix} \mathbf{c_1} \\ \mathbf{G_0 \otimes Ac}_2 + \mathbf{G_0 \otimes A_\Delta c}_2 + \mathbf{Iq}_2 \end{bmatrix}$$

with

$$\mathbf{RHS_M} = \begin{bmatrix} \mathbf{r_1} \\ \left(\mathbf{G_0 \otimes A + G_0 \otimes A_\Delta}\right)\mathbf{r}_2 \end{bmatrix},$$

where quantities $\mathbf{c}_1$ and $\mathbf{r}_1$ are now associated with all effects other than the additive.

Nonsymmetric Solvers

The presented system of equations is nonsymmetric and the matrix **H** may be semipositive definite. The PCG algorithm (Barrett et al., 1994) is applicable only to symmetric systems of equations. Therefore, it is important to find a suitable conjugate-gradient type algorithm and ensure that it would converge even with a poorly conditioned **H**. Barrett et al. (1994) and Van der Vorst (2003) reviewed and presented several algorithms for solving the linear systems of equations. Based on their studies, the standard algorithm for solving sparse systems with nonsymmetric LHS is bi-conjugate gradient stabilized (**Bi-CGSTAB**; Van der Vorst, 1992; see Appendix B). This algorithm requires 2 **LHS** times a vector products per round as opposed to just one with PCG. When that product uses the majority of the computing time, Bi-CGSTAB is about twice as expensive as PCG per round of iteration.

Choice of Preconditioners

In initial tests (results not reported), Bi-CGSTAB converged very quickly with the unsymmetric equations for small models, but not for large ones. This was traced to large off-diagonal elements of the unsymmetric equations. The standard way in conjugate-gradient types

of algorithms to improve convergence is by choice of a preconditioner **M**, which approximates **LHS** but is easily invertible (Van der Vorst, 2003). Then the system of equations solved is equivalent to

$$\mathbf{M}^{-1}\mathbf{LHS}\ \mathbf{w} = \mathbf{M}^{-1}\mathbf{RHS},$$

which has better numerical properties than the original system. The preconditioner is never used explicitly, but only in multiplications with a vector.

Assuming a diagonal preconditioner,

$$\mathbf{M}^{-1} = \mathrm{diag}\left(\mathbf{LHS}\right)^{-1} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix},$$

the **LHS** for regular equations after preconditioning is

$$\mathbf{M}^{-1}\ \mathbf{LHS} = \begin{bmatrix} \mathbf{D}_1\mathbf{X}'\mathbf{X} & \mathbf{D}_1\mathbf{X}'\mathbf{Z} \\ \mathbf{D}_2\mathbf{Z}'\mathbf{X} & \left[\mathbf{D}_2\left(\mathbf{Z}'\mathbf{Z} + \alpha\mathbf{A}^{-1}\right)\right] \end{bmatrix}.$$

The symmetry can be partially restored with a modified preconditioner:

$$\mathbf{M}_{\mathbf{M}}^{-1} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2\mathbf{A}^{-1} \end{bmatrix}.$$

Then

$$\begin{aligned} &\mathbf{M}_{\mathbf{M}}^{-1}\ \mathbf{LHS}_{\mathbf{M}} \\ &= \begin{bmatrix} \mathbf{D}_1\mathbf{X}'\mathbf{X} & \mathbf{D}_1\mathbf{X}'\mathbf{Z} \\ \mathbf{D}_2\left[\left(\mathbf{I} + \mathbf{A}^{-1}\mathbf{A}_{\Delta}\right)\mathbf{Z}'\mathbf{X}\right] & \mathbf{D}_2\left[\left(\mathbf{I} + \mathbf{A}^{-1}\mathbf{A}_{\Delta}\right)\mathbf{Z}'\mathbf{Z} + \alpha\mathbf{A}^{-1}\right] \end{bmatrix} \\ &= \mathbf{M}^{-1}\ \mathbf{LHS} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{D}_2\mathbf{A}^{-1}\mathbf{A}_{\Delta}\mathbf{Z}'\mathbf{X} & \mathbf{D}_2\mathbf{A}^{-1}\mathbf{A}_{\Delta}\mathbf{Z}'\mathbf{Z} \end{bmatrix}. \end{aligned}$$

When the genomic information is missing ($\mathbf{A}_{\Delta} = 0$), the preconditioned left-hand side of the unsymmetric system of equations is the same as with the preconditioned regular equations. With the genomic information, the off-diagonal elements are likely to be small for small $\mathbf{A}_{\Delta}$. The cost of the extra preconditioning is low because the product $\mathbf{D}_2\mathbf{A}^{-1}\mathbf{q}$, where $\mathbf{q}$ is a vector, can be done sequentially as $\mathbf{D}_2(\mathbf{A}^{-1}\mathbf{q})$.

Data

The data set included 10.5 million final scores on 6.2 million Holsteins as used for the recent genetic evaluation by the Holstein Association. Analyses were by a repeatability animal model. Two sets of mixed model equations were considered: regular and unsymmetric. For the second set, the genomic information was simulated for 5,000 randomly chosen animals as random numbers from the uniform distribution from 0 to b, where b was set to 0.0, 0.01, 0.03, and 0.05. For b = 0 there was no adjustment ($\mathbf{A}_{\Delta} = 0$). Only positive adjustments were included to avoid some elements of **H** being negative. Solving algorithms were PCG (for the regular equations only) and Bi-CGSTAB. The first algorithm used a diagonal preconditioner. The second algorithm used the modified preconditioner because no convergence was achieved with the diagonal preconditioner. In all cases, the stopping criterion was set at 10^{-12}. Computing was by the regular and modified program BLUP90IOD (Tsuruta et al., 2001) and was carried out on an Opteron system running at 3 GHz.

RESULTS AND DISCUSSION

The purpose of testing with the simulated genomic changes was to evaluate the computing feasibility of the method, and especially the robustness of the computing methodology. The results presented for the unsymmetric equations are only with the modified preconditioner. The Bi-CGSTAB diverged with the regular preconditioner and large data sets although it converged with small data sets. This is because products of **A** were very large for rows corresponding to popular bulls as all elements of **A** are positive; those products with $\mathbf{A}^{-1}$ are small because of cancellations; a contribution to a parent by a progeny in $\mathbf{A}^{-1}$ is proportional to [... 1.0 ... −0.5 ... −0.5 ...], which sums to 0.

Table 1 shows the number of rounds and computing time with PCG and Bi-CGSTAB for the regular and unsymmetric equations and with varying magnitudes of simulated changes. For the regular equations, Bi-CGSTAB was slightly faster but took twice the computing time (26 vs. 13 s). Figure 1 shows the convergence pattern for the regular equations. Whereas the pattern for PCG shows small fluctuations, the pattern for Bi-CGSTAB has more abrupt changes. Some differences in the number of rounds to convergence may be due to differences in the convergence criteria. However, the differences in solutions were very small (correlations >0.99999).

For the unsymmetric equations with no simulated changes, the number of rounds approximately doubled and the computing time increased by 30% (from 26 to 34 s). Adding small simulated changes (b = 0.01) increased the computing time per round by 10% (from 34 to 37 s) and slightly deteriorated convergence. The number of rounds increased by about 30% when changes were increased to b = 0.03 and again by 10% when

Table 1. The number of rounds (computing time per round in seconds) for different computing algorithms and different magnitudes of modification to the numerator relationship matrix

	Equation				
		Unsymmetric[2]			
Solving algorithm[1]	Regular	b = 0	b = 0.01	b = 0.03	b = 0.05
PCG	189 (13.1)	—	—	—	—
Bi-CGSTAB	166 (26.0)	318 (34.0)	369 (37.3)	477 (37.1)	520 (37.4)

[1]PCG = preconditioned conjugate gradient; Bi-CGSTAB = bi-conjugate gradient stabilized.
[2]Changes in relationships simulated from uniform (0, b) distribution for 5,000 randomly selected animals.

changes were increased by b = 0.05. Figure 2 shows the convergence pattern for the modified equations and b = 0.0 and 0.03. Much larger fluctuations than with the regular equations were observed, which may have been due to a more complex preconditioner. For a multiple-trait random regression model, Aguilar et al. (2008) observed much larger fluctuations in the convergence pattern with a block-diagonal preconditioner as compared with a diagonal one.

Additional computations will be necessary in practical applications of the method with the real genomic relationship matrix. For simple **H**, additional steps include the multiplications of $\mathbf{G}\text{-}\mathbf{A}_{22}$ and **A** by a vector. The last one can be done efficiently using the algorithm of Colleau (2002) in linear time (see Appendix A). The cost of this algorithm is equal to scanning the pedigree file twice and is small, especially with pedigrees in memory. Legarra et al. (2009) presented formulas for

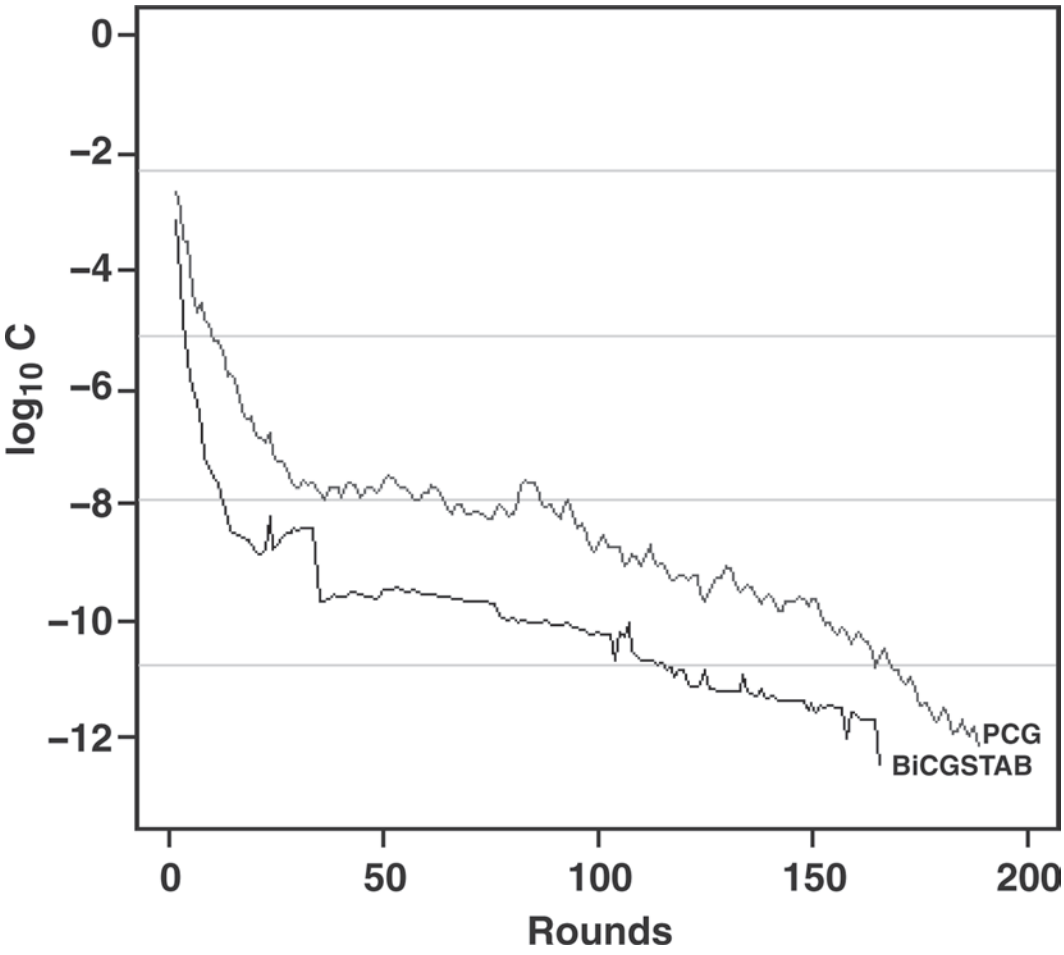

Figure 1. Convergence rate for the preconditioned conjugate gradient (PCG) and bi-conjugate gradient stabilized (Bi-CGSTAB) algorithms with the symmetric system of equations.

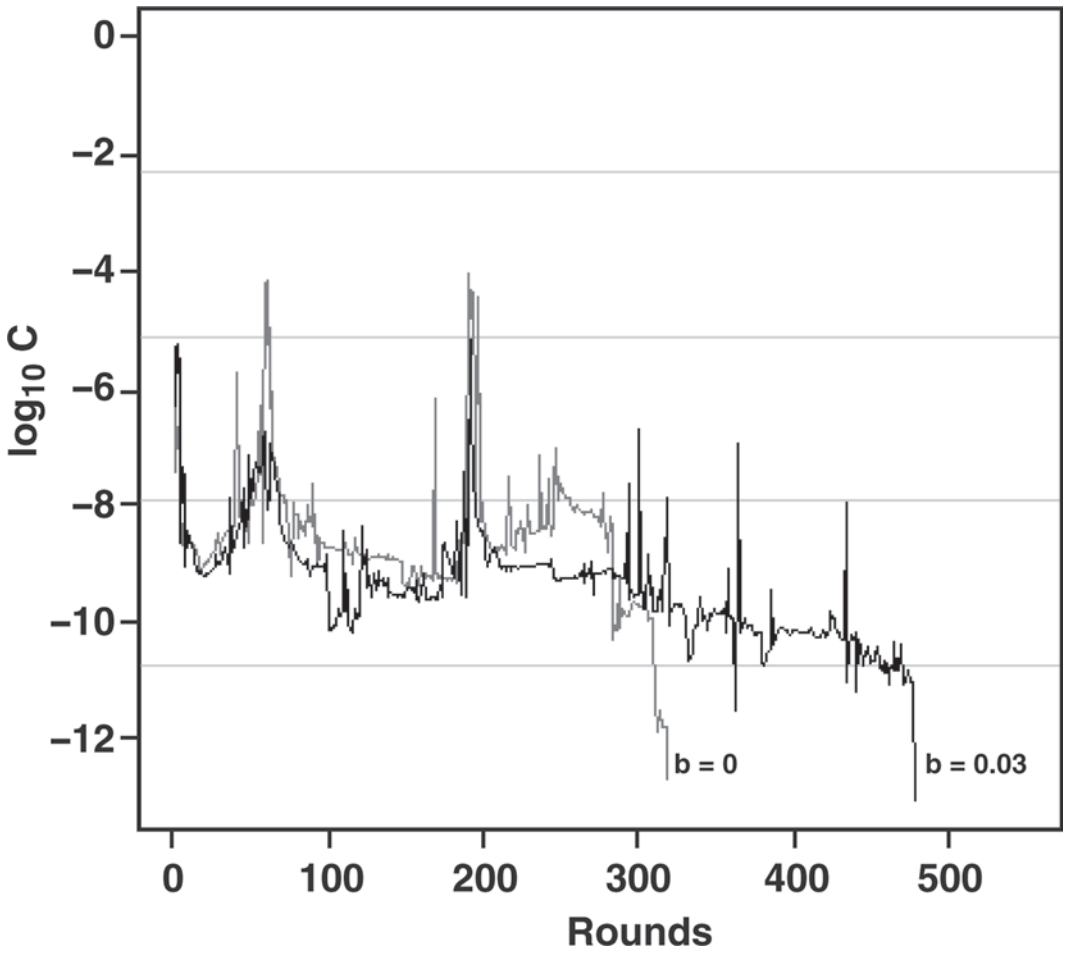

Figure 2. Convergence rate for the bi-conjugate gradient stabilized (Bi-CGSTAB) algorithm with the unsymmetric system of equations and either no (b = 0) or a middle level (b = 0.03) of simulated changes attributable to the genomic relationship matrix.

more realistic **H** and also computing details for a product of that **H** by a vector. With such a product, the only components that cannot be computed in linear but rather in quadratic time (matrix-vector multiplication) are those corresponding to **G** and possibly those due to $\mathbf{A}_{22}$. If $\mathbf{A}_{22}$ needs to be available explicitly, it can be computed by the method of Aguilar and Misztal (2008). When applied to 17 million Holsteins, that method calculated about 80,000 inbreeding coefficients/s. Assuming that computing one relationship costs no more than computing one inbreeding coefficient, on average, the computation of $\mathbf{A}_{22}$ for 20,000 animals would take 40 min. Alternatively, $\mathbf{A}_{22}$ can be computed by the repeated applications of the algorithm of Colleau (2002), in which the vector to multiply by would contain one 1.0 and zeros elsewhere.

When the number of genotyped animals is very high, say >50,000, storage and computations with matrix **G** and possibly $\mathbf{A}_{22}$ can be quite involving. A few choices may be applicable. First, some computations may easily be done in parallel. Current computers routinely include 4 processors (cores) per processor module, and computers with 4 modules are readily available. Second, some elements in $\mathbf{A}_{\Delta}$ may be very small or unimportant and could be neglected. Neglecting small elements in the computation of sparse inverse for the purpose of calculating accuracies reduced the computations by 50 times while retaining high precision (Thompson et al., 1994). Finally, genotypes of some animals may be unimportant and do not have to be included.

In summary, we have demonstrated that mixed model equations with small modifications to the numerator relationship matrix can be solved efficiently by conjugate-gradient type algorithms. Only a few modifications may be required for existing programs using the PCG algorithm.

ACKNOWLEDGMENTS

Discussions with Rohan Fernando (Department of Animal Science, Iowa State University, Ames), Jeff O'Connell (University of Maryland School of Medicine,

Baltimore), Paul VanRaden (Animal Improvement Programs Laboratory, ARS, USDA, Beltsville, MD), Curt Van Tassel (Bovine Functional Genomics Laboratory, ARS, USDA, Beltsville, MD), and Bruce Tier (Animal Genetics and Breeding Unit, University of New England, Armidale, Australia) are gratefully acknowledged. Also acknowledged are the encouragement to pursue this study by Tom Lawlor and the financial support by the Holstein Association, Brattleboro, Vermont (IM, IA), the EADGENE network of excellence, Agence National de la Recherche project AMASGEN, and Maison de Relations Internationales (INRA, France). We also appreciate the very diligent work by the 2 anonymous reviewers.

REFERENCES

Aguilar, I., and I. Misztal. 2008. Recursive algorithm for inbreeding coefficients assuming non-zero inbreeding of unknown parents. J. Dairy Sci. 91:1669–1672.

Aguilar, I., S. Tsuruta, and I. Misztal. 2008. Computing options for multiple trait test day random regression models with account of heat tolerance and national datasets. J. Dairy Sci. 91(Suppl. 1):9. (Abstr.)

Barrett, R., M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. 1994. Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. SIAM, Philadelphia, PA.

Cole, J. B., P. M. VanRaden, J. R. O'Connell, C. P. Van Tassell, G. R. Wiggans, T. S. Sonstegard, R. D. Schnabel, J. F. Taylor, and G. R. Wiggans. 2009. Distribution and location of genetic effects for dairy traits. J. Dairy Sci. 92:2931–2946.

Colleau, J. J. 2002. An indirect approach to the extensive calculation of relationship coefficients. Genet. Sel. Evol. 34:409–421.

Gianola, D., R. L. Fernando, and A. Stella. 2006. Genomic-assisted prediction of genetic value with semiparametric procedures. Genetics 173:1761–1776.

Guillaume, F., S. Fritz, D. Boichard, and T. Druet. 2008. Short communication: correlations of marker-assisted breeding values with progeny-test breeding values for eight hundred ninety-nine French Holstein bulls. J. Dairy Sci. 91:2520–2522.

Harville, D. A. 1976. Extension of the Gauss-Markov theorem to include the estimation of random effects. Ann. Stat. 4:384–395.

Henderson, C. R. 1976. A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values. Biometrics 32:69–83.

Henderson, C. R. 1984. Applications of Linear Models in Animal Breeding. Univ. Guelph, Guelph, Ontario, Canada.

Henderson, C. R. 1985. Best linear unbiased prediction using relationship matrices derived from selected base populations. J. Dairy Sci. 68:443–448.

Legarra, A., I. Aguilar, and I. Misztal. 2009. A relationship matrix including full pedigree and genomic information. J. Dairy Sci. 92:4656–4663.

Legarra, A., and I. Misztal. 2008. Technical note: Computing strategies in genome-wide selection. J. Dairy Sci. 91:360–366.

Meuwissen, T. H. E., B. J. Hayes, and M. E. Goddard. 2001. Prediction of total genetic value using genome-wide dense marker maps. Genetics 157:1819–1829.

Neuner, S., R. Emmerling, G. Thaller, and K.-U. Götz. 2008. Strategies for estimating genetic parameters in marker-assisted best linear unbiased predictor models in dairy cattle. J. Dairy Sci. 91:4344–4354.

Quaas, R. L. 1988. Additive genetic model with groups and relationships. J. Dairy Sci. 71:1338–1345.

Thompson, R., N. R. Wray, and R. E. Crump. 1994. Calculation of prediction error variances using sparse matrix methods. J. Anim. Breed. Genet. 111:102–109.

Tsuruta, S., I. Misztal, and I. Stranden. 2001. Use of the preconditioned conjugate gradient algorithm as a generic solver for mixed-model equations in animal breeding applications. J. Anim. Sci. 79:1166–1172.

Van der Vorst, H. 1992. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. 13:631–644.

Van der Vorst, H. 2003. Iterative Krylov Methods for Large Linear Systems. Cambridge University Press, Cambridge, UK

VanRaden, P. M. 2008. Efficient methods to compute genomic predictions. J. Dairy Sci. 91:4414–4423.

VanRaden, P. M., C. P. Van Tassell, G. R. Wiggans, T. S. Sonstegard, R. D. Schnabel, J. F. Taylor, and F. S. Schenkel. 2009. Invited Review: Reliability of genomic predictions for North American Holstein bulls. J. Dairy Sci. 92:16–24.

APPENDIX A

Below we show how to create a product $\mathbf{Aq}$, where $\mathbf{A}$ is the numerator relationship matrix and $\mathbf{q}$ is a vector. The recurrence equation for the additive effect is

$$\mathbf{a} = \mathbf{Pa} + \phi,$$

where $\mathbf{a}$ is a vector of animals ordered from oldest to youngest, ϕ is a diagonal matrix of Mendelian samplings, and $\mathbf{P}$ is a matrix relating animals to their parents; this matrix has at most 2 elements per row, both equal to 0.5. Following Quaas (1988),

$$\text{Var}(\mathbf{a}) = \mathbf{A} = (\mathbf{I} - \mathbf{P})^{-1}\mathbf{D}(\mathbf{I} - \mathbf{P})^{-1'},$$

where $\mathbf{D} = \text{var}(\phi)$. Colleau (2002) showed that the product of $\mathbf{A}$ by a vector, for example,

$$\mathbf{v} = \mathbf{Aq} = (\mathbf{I} - \mathbf{P})^{-1}\mathbf{D}(\mathbf{I} - \mathbf{P})^{-1'}\mathbf{q} = (\mathbf{I} - \mathbf{P})^{-1}\mathbf{D}[(\mathbf{I} - \mathbf{P})^{-1'}\mathbf{q}],$$

can be solved in linear time. In particular, quantities $\mathbf{r} = (\mathbf{I} - \mathbf{P})^{-1'}\mathbf{q}$ and $\mathbf{v} = (\mathbf{I} - \mathbf{P})^{-1}\mathbf{Dr}$ can be obtained by solving $(\mathbf{I} - \mathbf{P})'\mathbf{r} = \mathbf{q}$ and $(\mathbf{I} - \mathbf{P})\mathbf{v} = \mathbf{Dr}$, each one in a single sweep because $(\mathbf{I} - \mathbf{P})$ is triangular. The scalar formulas are

$$r_i = r_i + q_i;\ r_{si} = r_{si} + r_i/2;\ r_{di} = r_{di} + r_i/2;\ i = n, ..., 1$$

$$v_i = d_i r_i + (v_{si} + v_{di})/2,\ i = 1, ..., n,$$

where s_i and d_i are positions of the sire and dam of animal i, respectively.

The Colleau (2002) algorithm can be used to compute products of sections of matrices. For instance, the products below show how to compute $\mathbf{A}_{12}\mathbf{q}$, $\mathbf{A}_{22}\mathbf{q}$, $\mathbf{A}_{21}\mathbf{q}$, or $\mathbf{A}_{22}\mathbf{q}$:

$$\begin{bmatrix}\mathbf{A_{11}} & \mathbf{A_{12}}\\ \mathbf{A_{21}} & \mathbf{A_{22}}\end{bmatrix}\begin{bmatrix}\mathbf{q}\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{A_{11}q}\\ \mathbf{A_{21}q}\end{bmatrix},\ \begin{bmatrix}\mathbf{A_{11}} & \mathbf{A_{12}}\\ \mathbf{A_{21}} & \mathbf{A_{22}}\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \mathbf{q}\end{bmatrix} = \begin{bmatrix}\mathbf{A_{12}q}\\ \mathbf{A_{22}q}\end{bmatrix}.$$

APPENDIX B

The pseudo-program below implements the Bi-CG-STAB (Van der Vorst, 1992) for a system of equations $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{M}$ being a preconditioner. The major expenses in the algorithm are products of $\mathbf{A}$ by a vector, possibly followed by products of $\mathbf{M}^{-1}$, but only if $\mathbf{M}$ is of complex structure.

Compute $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$ for some initial guess $\mathbf{x}^{(0)}$

Choose $\tilde{\mathbf{r}}$ (for example, $\tilde{\mathbf{r}} = \mathbf{r}^{(0)}$)

for i = 1, 2, …

$\rho_{i-1} = \tilde{\mathbf{r}}'\mathbf{r}^{(i-1)}$

if ρ_{i-1} = 0 method fails

if i = 1

$\mathbf{p}^{(i)} = \mathbf{r}^{(i-1)}$

else

$\beta_{i-1} = \left(\frac{\rho_{i-1}}{\rho_{i-2}}\right)\left(\frac{\alpha_{i-1}}{\omega_{i-1}}\right)$

$\mathbf{p}^{(i)} = \mathbf{r}^{(i-1)} + \beta_{i-1}\left(\mathbf{p}^{(i-1)} - \omega_{i-1}\boldsymbol{\upsilon}^{(i-1)}\right)$

endif

solve $\boldsymbol{M}^{-1}\hat{\mathbf{p}} = \mathbf{p}^{(i)}$

$\upsilon^{(i)} = \mathbf{A}\hat{\mathbf{p}}$

$\alpha_i = \frac{\rho_{i-1}}{\tilde{\mathbf{r}}'\boldsymbol{\upsilon}^{(1)}}$

$\mathbf{g} = \mathbf{r}^{(i-1)} - \alpha_i\boldsymbol{\upsilon}^{(i)}$

check norm of g; if small enough: set

$\mathbf{x}^{(i)} = \mathbf{x}^{(i-1)} + \alpha_i\hat{\mathbf{p}}$ and stop

solve $\mathbf{M}\hat{\mathbf{g}} = \mathbf{g}$

$\mathbf{t} = \mathbf{A}\hat{\mathbf{g}}$

$\omega_i = \frac{\mathbf{t}'\mathbf{g}}{\mathbf{t}'\mathbf{t}}$

$\mathbf{x}^{(i)} = \mathbf{x}^{(i-1)} + \alpha_i\hat{\mathbf{p}} + \omega_i\hat{\mathbf{g}}$

$\mathbf{r}^{(i)} = \mathbf{g} - \omega_i\mathbf{t}$

check for convergence; continue if necessary

for continuation it is necessary that $\omega_i \neq 0$

end