February 1982

Computer Algorithm for the Recursive Method of Calculating Large Numerator Relationship Matrices

G. F. S. Hudson
Cornell University

R. L. Quaas
Cornell University

L. Dale Van Vleck
University of Nebraska-Lincoln, dvan-vleck1@unl.edu

Computer Algorithm for the Recursive Method of Calculating Large Numerator Relationship Matrices

G.F.S. HUDSON, R. L. QUAAS, and L. D. VAN VLECK
Department of Animal Science
Cornell University
Ithaca, NY 14853

ABSTRACT

A method of storing and retrieving nonzero elements of a large, sparse symmetric matrix using only internal computer storage is presented. An algorithm for applying the recursive or tabular method of calculating the numerator relationship matrix of a large group of animals is given. The algorithm is applicable to inbred and noninbred populations and can be used on sire/dam or sire/maternal grandsire pedigrees.

INTRODUCTION

Wright's (9) coefficients of relationship and inbreeding are calculated most easily by the well-known recursive or tabular method attributed to J. L. Lush by Emik and Terrill (1). Henderson (4) and Quaas (6) have developed methods for computing the inverse of the numerator relationship matrix required for best linear unbiased predictions (BLUP) of breeding values (3) without calculating and inverting the matrix directly. However, knowledge of relationships among a group of animals is useful for examining the effect of, and preventing, inbreeding. Furthermore, the numerator relationship matrix, **A** say, is required if relationships among animals are incorporated into variance component estimation models (8).

This paper describes an algorithm for storing and computing **A** for large groups of animals. The method is explained by the hypothetical example of Table 1 in which animals have been numbered consecutively in decreasing age order. The corresponding relationship matrix is in Table 2.

STORING NONZERO ELEMENTS OF A

Suppose **A** has order N^2 and contains n nonzero elements in the upper half (i.e., on and above the diagonal). The ijth element, a_{ij}, is the numerator relationship between animals i and j. The entire upper half could be stored in a single array of dimension $N(N + 1)/2$, but computer storage space becomes limiting when N is more than a few hundred. The method presented requires storing only nonzero elements as half-word integers in an array of dimension n. Two half-word integer arrays, of dimension n and N, store information necessary to recover a given element of the original matrix **A**. On the IBM 370 computer a full-word (or long) integer requires 32 bits of memory whereas a half-word (or short) integer uses only 16 bits of memory. The capability to use half-word integers is dependent on the computer available, but the principles are applied easily to any system.

To reduce storage requirements and increase the order of the matrix that can be calculated, relationship coefficients are expressed as integers by multiplication by a large power of 2, e.g., 2^{14}. The integers can be stored in a half-word integer array rather than the full-word floating-point array necessary to store a decimal fraction. For example, relationships 1/2, 3/8, 5/16 are stored as 8192, 6144, and 5120. Other coefficients are expressed in a similar manner. For clarity in this paper, relationships are left in more familiar decimal form.

The storage method is similar to Scheme II of (7). Nonzero elements of the upper-half of **A** are stored in rows: in one array, COEFF, of dimension n (Table 3). The column subscripts of off-diagonal elements of **A** are stored in a half-word integer array of dimension n, COL. For example, the 11th location in COEFF contains an element of the 5th column of **A**. Locations of diagonal elements in COEFF are stored in a half-word integer array, ROW, of dimension N. For example, the location of the 4th diagonal in COEFF is stored in the 4th entry of ROW. Table 3 shows the contents of the arrays after all calculations have been

Received February 26, 1982.

TABLE 1. Example pedigree.

Animal	Sire	Dam
1	...[a]	...
2	...	...
3	...	...
4	1	3
5	2	3
6	2	...
7	2	6

[a]Missing information.

completed. At the beginning of the computations n is unknown, so the necessary length of COEFF and COL has to be estimated.

Recovery of any particular element of **A** is accomplished by searching for the column subscript in the interval of COL defined by the diagonal elements of the current and next rows. Within each row, the elements of COL are in strictly increasing order; thus, a bisection search can be used to determine the location or absence of the required column subscript. Efficient search algorithms are in (5), but an outline of a bisection search follows. The interval of COL that contains the column subscripts of the appropriate row is bisected to find the approximate midpoint of the interval. The entry of COL at the midpoint is compared with the search argument, and the upper or lower half of the interval is taken as a new interval depending on whether the midpoint is less than or greater than the search argument. The process continues until the required entry is found at the midpoint or until the interval collapses in which case the search argument is not in the original interval. If an element below the diagonal is required, e.g., $a_{i,j}$ with $i > j$, the subscripts must be interchanged because only the upper half of **A** is stored. Recovery of diagonal elements requires only the location in ROW, e.g., $a_{3,3}$ is known to be stored in COEFF (7) as ROW (3) contains 7.

PROCEDURE FOR CALCULATING NONZERO ELEMENTS OF A

Animals must be identified in decreasing age order with sequential numbers from 1 to N, so that 1 identifies the oldest animal and N the youngest. Sire and dam identification for each animal are stored in two half-word integer arrays each of dimension N. A zero is stored if parental information is missing.

For each row of **A**, the diagonal element is calculated first, followed by all off-diagonals to the right of the diagonal. In the last row there are no off-diagonals, but the diagonal has to be computed for completeness of the matrix. Explanation of the algorithm is simplier if we assume that the first $i - 1$ rows have already been computed and stored in the manner described in the previous section.

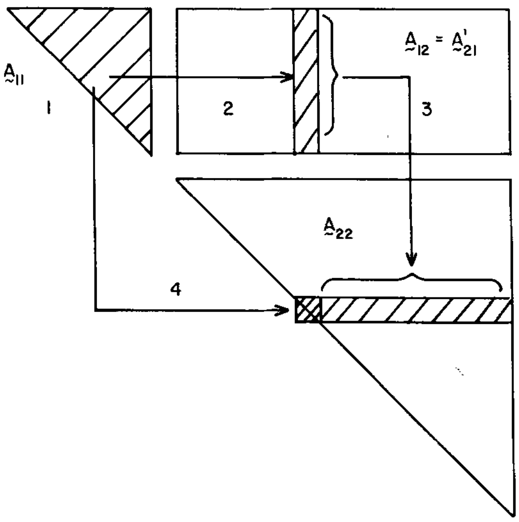
The diagonal element of the ith row, a_{ii}, is $1 + f_i$, where $f_i = 1/2\ a_{jk}$ is the inbreeding coefficient of the ith animal and j and k are the sire and dam of the ith animal (not necessarily respectively), with $k > j$ for the search procedure. If either sire or dam identification is missing or if $a_{jk} = 0$, then $a_{ii} = 1$, except as indicated below where only sire and maternal grandsire are identified. The a_{ii} is stored in the next available entry of COEFF, and its position stored in the ith entry of ROW. (A counter is needed to keep track of the number of filled entries in COL and COEFF.)

TABLE 2. Numerator relationship matrix for the example of Table 1.

Animal	1	2	3	4	5	6	7
1	1	0	0	1/2	0	0	0
2	0	1	0	0	1/2	1/2	3/4
3	0	0	1	1/2	1/2	0	0
4	1/2	0	1/2	1	1/4	0	0
5	0	1/2	1/2	1/4	1	1/4	3/8
6	0	1/2	0	0	1/4	1	3/4
7	0	3/4	0	0	3/8	3/4	1+1/4

TABLE 3. Contents of arrays necessary for storing A for the example after completion of all calculations.

Index of location in array	COEFF	COL	ROW
1	1	0	1
2	.5	4	3
3	1	0	7
4	.5	5	10
5	.5	6	12
6	.75	7	15
7	1	0	17
8	.5	4	
9	.5	5	
10	1	0	
11	.25	5	
12	1	0	
13	.25	6	
14	.375	7	
15	1	0	
16	.75	7	
17	1.25	0	



1 Calculate and store A_{11} in core.
2 Calculate a column of A_{12} from A_{11}.
3 Use the column of A_{12} to calculate the off-diagonals of the corresponding row of A_{22}.
4 Calculate the diagonal of A_{22} from A_{11}.
5 Write the column of A_{12} and row of A_{22} on external storage.

Figure 1. Schematic representation of procedure for calculating partitioned relationship matrix.

Off-diagonals, a_{ij} ($j = i + 1, i + 2, \ldots, N$) are calculated by $1/2\ (a_{jk} + a_{jl})$ where k and l are sire and dam of the jth animal. Either k or l may equal i, in which case the ith diagonal is required. If $i > k$ (or $i > l$) the search is for a_{ki} (or a_{li}) in the ith row, and the upper limit of the interval for the bisection search is the last entry of COL containing a column subscript, as indicated by the counter. The calculated value of a_{ij} then is stored in the next available entry of COEFF, the column subscript, j is stored in the corresponding entry of COL and the counter incremented by 1.

When only sire and maternal grandsire (mgs) identifications are available, the required calculations are $f_i = 1/4\ a_{k,l}$, with k and l the sire and mgs of i, and $a_{i,j} = 1/2\ a_{i,p} + 1/4\ a_{i,q}$, with p and q the sire and mgs, respectively, of j.

REDUCING COMPUTER STORAGE REQUIREMENTS

Even when only nonzero elements of the upper half of A are stored, computer capacity may be exceeded by large matrices. Storage requirements can be reduced further as follows. Order the list of animals so that the last $N - k$ in the list have no descendants, and partition A correspondingly:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

with A_{11} of order k^2, A_{12} of order $k \times (N - k)$, A_{22} of order $(N - k)^2$, and A_{21} equals the transpose of A_{12}. Relationship coefficients in A_{12} are dependent only on those in A_{11} and not on any others in A_{12} because the animals represented by the $N - k$ columns of A_{12} are not descendants of any other animals represented by columns of A_{12}. Similarly, elements of A_{22} are functions only of the elements of A_{11} or A_{21} and not of the other elements of A_{22}. This partitioning allows A to be calculated as follows:

1 calculate A_{11} as described in the previous section,
2 for each column of A_{12}, a_j, $j = N - k, N - k + 1, \ldots, N$
   i) calculate a_j from elements of A_{11}
   ii) use a_j as a row of A_{21} to calculate the off-diagonals to the right of the diagonal in the corresponding row of A_{22}

TABLE 4. Number of bulls and nonzero relationships and computer[a] time required to generate the relationship matrix in five breeds of dairy cattle (based on sire/maternal grandsire pedigrees).

Breed	Number of bulls	Number (percent) of nonzero coefficients in upper half of A	Computer[a] time (s)
Ayrshire	192	1,761 (9.5)	7.5
Guernsey	493	8,028 (6.6)	67
Holstein	4094	1,828,089 (21.8)	. . .[b]
Jersey	611	18,878 (10.1)	130
Brown Swiss	204	4,734 (22.6)	12

[a]IBM 370/138 (memory capacity 1200K).

[b]See text.

iii) calculate the diagonal of A_{22} from elements of A_{11}

iv) write on external storage (tape or disk) a_j and the row of A_{22} as they are calculated.

The procedure for calculating the partitioned form of **A** is shown schematically in Figure 1. Storage requirement is reduced to that necessary for nonzero elements of the upper half of A_{11} and one column of A_{12} (row of A_{21}). This strategy depends on A_{11} being small relative to **A** or on A_{11} having only a small proportion of the total of nonzero elements in **A**.

The relationship matrix of a cow population is probably too large to compute in the manner described here. However, most relationships of the female side of pedigrees are likely to be within herds, and a separate relationship matrix for each herd easily could be calculated. Relationships among bulls first would have to be computed separately and incorporated into within herd matrices as required.

APPLICATIONS

Van Vleck and Hudson (8) used numerator relationship matrices to adjust estimates of heritability for relationships among sires. In the five dairy cattle populations they studied, sires were not sufficiently related to each other to cause a major change in heritability estimates compared with estimation ignoring relationships. Grimes and Harvey (2) also present a method of estimating genetic variances and covariances that incorporates relationships among individuals.

Knowledge of relationships among bulls allows calculation of inbreeding of cows if sires and maternal grandsires of the cows are known. Effects of inbreeding on traits of economic importance then can be studied. Inbreeding can be prevented by calculating the inbreeding coefficient of future progeny.

Table 4 shows the number of nonzero coefficients and the time required to compute the relationship matrix for sires of four breeds with sire and mgs identification. Nonzero elements represented 7 to 23% of the upper half of the matrix.

The relationship matrix for Holsteins was too large to store in core and was calculated from the partitioned matrix method described in section 4. Because of the large number of coefficients, the entries stored in ROW exceeded the upper bound for half-word integers, so ROW was changed to a full-word integer array for Holsteins. A_{11} was of order 1503 and took approximately 20 min to generate on an IBM 370/138. The complete matrix required 125 min computer time, but this included writing elements of **A** on tape. The times shown in Table 4 for other breeds exclude input and output operations.

If only inbreeding coefficients are required, the algorithm of Quaas (6) may be more rapid than that presented here because no search is required. However, his method is unsuitable for generating the complete matrix for a large group of animals, which is required for calculation of future inbreeding coefficients.

ACKNOWLEDGMENTS

This research was supported in part by a grant from Eastern Artificial Insemination Cooperative, Inc., Ithaca, NY.

REFERENCES

1 Emik, L. O., and C. R. Terrill. 1949. Systematic procedures for calculating inbreeding coefficients. J. Hered. 40:51.

2 Grimes, L. W., and W. R. Harvey. 1980. Estimation of genetic variances and co-variances using symmetric differences squared. J. Anim. Sci. 50:634.

3 Henderson, C. R. 1973. Sire evaluation and genetic trends. Pages 10–41 *in* Proc. Anim. Breeding Genet. Symp. in Honor of Dr. Jay L. Lush. Am. Soc. Anim. Sci. Am. Dairy Sci. Assoc., Champaign, IL.

4 Henderson, C. R. 1976. A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values. Biometrics 32:69–88.

5 Knuth, D. E. 1973. The art of computer programming. Vol. 3. Sorting and Searching. Addison-Wesley, Reading, MA.

6 Quaas, R. L. 1976. Computing the diagonal elements and inverse of a large numerator relationship matrix. Biometrics 32:949.

7 Tewerson R. P. 1973. Sparse matrices. Academic Press, New York, NY.

8 Van Vleck, L. D., and G. F. S. Hudson. 1982. Relationships among sires in estimating genetic variance. J. Dairy Sci. 65:1663.

9 Wright, S. 1922. Coefficients of inbreeding and relationships. Am. Nat. 56:330.