Indirect Solution of Mixed Model Equations

I. MISZTAL and D. GIANOLA
Department of Animal Sciences
University of Illinois
Urbana 61801

ABSTRACT

Large scale genetic evaluation of animals by best linear unbiased prediction can have a high computational cost. This is partly due to the need to set up mixed model equations, which are then solved in an iterative way. Solutions can also be obtained by successive averaging without setting the mixed model equations directly. Formulae are presented for a class of models with fixed and random factors, including an additive relationship matrix. Two iterative procedures were investigated, Gauss-Seidel and Jacobi. With a balanced data set, putting restrictions on fixed effects is not effective for improving convergence rates in Gauss-Seidel but is essential in Jacobi. Computational techniques needed to implement the indirect procedures are discussed.

INTRODUCTION

An approach often used for solving large mixed model equations involves setting the system of equations explicitly before iteration (7). This requires complex programming, large temporary space, and long running time. Schaeffer and Kennedy (4) showed how solutions can be obtained without setting up the system of equations. Their strategy, named indirect approach, performs Gauss-Seidel (G-S) or successive overrelaxation (SOR) iteration while reading the data files rather than the matrix of coefficients. The data files, read once per round of iteration, consist of two copies of an observation file, each sorted for a different factor with a large number of levels, and three copies of a relationship file, also sorted differently. This G-S indirect approach was found by Schaeffer and Kennedy (5) to be much faster than the “traditional” approach, and programming was greatly simplified. Although the indirect approach is based on mixed model equations, these are not used explicitly.

Misztal (2, 3) proposed another indirect approach using Jacobi (J) iteration (8). Programming is simpler, because only one unsorted copy of each of the observation and relationship files need to be read per round of iteration. Basic J iteration has a poor convergence rate, especially for models with many factors or with a random factor including the numerator relationship matrix. However, putting restrictions on the effects of levels of fixed factors and extending the method to second-order J made it competitive in running time relative to other methods; convergence rate was approximately half that of SOR (8). When G-S was used for one large factor and second-order J for the other, convergence rate was faster and approached the speed of SOR when an animal model was used (8). This “hybrid” procedure required reading both the observation and relationship files, but the former had to be sorted by the large factor upon which G-S operated.

This paper presents the indirect approach in terms of successive averaging, investigates its properties, and describes several computing options.

METHODS

Let the mixed model equations be represented as:

$$Ab = r$$

where A is a p x p matrix of coefficients, b is a vector of unknowns, and r is a vector of right-hand sides (RHS). The iterative method of SOR can be described (8) as:

$$b^{(n+1)}_i = b^{(n)}_i + \alpha (r_i - \sum_{j=1}^{i-1} a_{ij} b^{(n+1)}_j - \sum_{j=1}^{p} a_{ij} b^{(n)}_j) / a_{ii}$$

2

Received October 6, 1986.
Accepted December 15, 1986.
1 On leave from Warsaw Agricultural University (SGGW-AR), Przejazd 4, 05-840 Brwinow, Poland.

1987 J Dairy Sci 70:716–723 716
where $b_i$ and $r_i$ are elements of $b$ and $r$, respectively, $a_{ij}$ and $a_{ij}$ are elements of $A$, $n$ is iterate number, and $\alpha$ is a relaxation factor. For $\alpha = 1$, SOR becomes G-S.

The J method does not use updated solutions during the same round of iteration. The second-order J iterative procedure (8) is:

$$b_i^{[n+1]} = b_i^{[n]} + \alpha(b_i^{[n]} - b_i^{[n-1]})$$

$$+ (r_i - \sum_{j=1}^{p} a_{ij} b_j^{[n]})/a_{ii}$$

or, in matrix form:

$$b^{[n+1]} = b^{[n]} + \alpha(b^{[n]} - b^{[n-1]})$$

$$+ D^{-1} (r - A b^{[n]})$$

where $D$ is a matrix containing the diagonal elements of $A$. Second-order J becomes regular J for $\alpha = 0$.

Consider the mixed linear model:

$$Y_{ijkl} = h_i + g_j + s_k + e_{ijkl}$$

where $h_i$ and $g_j$ are fixed effects and $s_k$ and $e_{ijkl}$ are random effects with null means and covariance matrices:

$$V = \begin{bmatrix} \sigma^2_h & 0 & 0 \\ 0 & \sigma^2_g & 0 \\ 0 & 0 & \sigma^2_s \end{bmatrix}$$

The $h$, $g$, and $s$ effects may correspond to herd, group, and sire effects, respectively. Each observation $(Y_{ijkl}, h_i, g_j, s_k)$ gives the following "contributions" to the mixed model equations:

<table>
<thead>
<tr>
<th>Coefficient matrix</th>
<th>Right-hand side</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_i$</td>
<td>$g_j$</td>
</tr>
<tr>
<td>$h_i$</td>
<td>1</td>
</tr>
<tr>
<td>$g_j$</td>
<td>1</td>
</tr>
<tr>
<td>$s_k$</td>
<td>1</td>
</tr>
</tbody>
</table>

Thus, the "contributions" of an observation to the left-hand side (LHS) and RHS of [1] are $h_i + g_j + s_k$ and $Y_{ijkl}$, respectively. For all observations, the LHS must equal the RHS for each level of each factor. The equation for $h_i$ is then:

$$\sum_{jkl} (h_i + g_j + s_k) = \sum_{jkl} Y_{ijkl}$$

To solve this $h_i$ equation by G-S as in [2] with $\alpha = 1$, we form the expression:

$$h_i = h_i + (RHS - LHS)/n_{hi}$$

$$= h_i + [\sum_{jkl} (Y_{ijkl} - h_i - g_j - s_k)]/n_{hi}$$

$$= \sum_{jkl} (Y_{ijkl} - g_j - s_k)/n_{hi}$$

$$= (Y_{ijkl} - g_j - s_k)/n_{hi}$$

where $n_{hi}$ is the number of observations in level $i$ of factor $h$. Similar formulas for the remaining factors are:

$$g_j = (Y_{ijkl} - h_i - s_k)$$

and:

$$s_k = \frac{n_{sk}}{n_{sk} + \lambda} (Y_{ijkl} - h_i - g_j)$$

where $n_{sk}$ is the number of observations in level $k$ of random factor $s$ and $\lambda = \sigma^2_s/\sigma^2_s$. A round of G-S iteration consists of sequential evaluation of [7], [8], and [9]. If [7], [8], and [9] are evaluated simultaneously, this gives one round of J iteration.

**PROPERTIES OF GAUSS-SEIDEL AND JACOBI ITERATION**

Using a balanced data set, we assume that the superiority of a given method when data are balanced also holds with unbalanced layouts. For example, Thompson and Meyer (6) studied an algorithm for restricted maximum likelihood estimation of variance components and found that a modification that led to faster convergence with balanced data had a similar effect with unequal numbers.

In a balanced layout, let $H = \text{number of herds}$, $G = \text{number of groups}$, $S = \text{number of sires}$, and $N = \text{number of records in a herd \times group \times sire subclass}$. As all effects in a balanced layout are equally distributed, [7] can be written:

where $\bar{y}_{i..}$ is the average of all records in herd $i$, and $g$ and $s$ are the averages of the group and sire effects, respectively. Then:

$$h_i = \bar{y}_{i..} - g - s$$ \[10\]

Now, \[8\] can be written as:

$$g_j = \bar{y}_{j..} - h - s$$ \[11\]

where $\bar{y}_{j..} = y_{j..} /N_{j..}$ and $h = \Sigma h_i /H$.

Equation \[9\] becomes:

$$s_k = m(\bar{y}_{..k} - h - g)$$ \[12\]

where $m = n/(n + \lambda)$, $n$ is progeny group size, and $\bar{y}_{..k} = y_{..k} /N_{..k}$.

Gauss-Seidel Iteration

Assuming that the starting values of $g_j$ and $s_k$ are null, \[10\] can be written as:

$$h_i = \bar{y}_{i..}$$ \[13\]

Decompose $h_i$ into a mean and a restricted effect:

$$h_i = \mu + h_i^*$$

$$\Sigma h_i^* = 0$$ \[14\]

Then the solution for group $j$ in \[11\] can be written as:

$$g_j = \bar{y}_{j..} - h = \bar{y}_{j..} - \Sigma(\mu + h_i^*)/H = \bar{y}_{j..} - \mu$$ \[15\]

Now, from \[14\], $\mu = h_i - h_i^*$, so write:

$$\mu = \Sigma(h_i - h_i^*)/H = \Sigma h_i /H = \bar{h}$$ \[16\]

Using \[13\] in \[16\], the “current” value of $\mu$ is:

$$\mu = \Sigma \bar{y}_{i..} /H = y_{..}/NHG$$

so that the group solution \[15\] in round 1 of G-S becomes:

$$g_j = \bar{y}_{j..} - \bar{y}_{..}$$ \[17\]

Thus:

$$\Sigma g_j = \Sigma(\bar{y}_{j..} - \bar{y}_{..}) = 0$$

Finally, using \[13\] and \[17\] in \[12\], the sire solutions in round 1 of G-S is:

$$s_k = m(\bar{y}_{..k} - \bar{h} - g) = m(\bar{y}_{..k} - \bar{y}_{..})$$ \[18\]

At the first round, the sum of the sire solution is null as:

$$\Sigma s_k = m \Sigma(\bar{y}_{..k} - \bar{y}_{..}) = 0$$

In the next round of iteration, i.e., applying \[10\], \[11\], and \[12\] again with \[13\], \[17\], and \[18\] would give the same solutions, indicating that convergence has occurred. Because of the parametrization chosen, the solutions for the first factor (herd) include a mean, and those for other effects are “mean-free”. If the first computed factor were random, the solutions would not be correct, as such a factor cannot contain the mean.

Jacobi Iteration

In the J method, expressions \[10\], \[11\], and \[12\] are evaluated simultaneously. Assuming again that the starting values are null, we have:

$$h_i = \bar{y}_{i..} = h_i^{GS}$$ \[19\]

so the first round solution for $h_i$ is the same as in G-S. Likewise:

$$g_j = \bar{y}_{j..} = \bar{y}_{..} + (\bar{y}_{j..} - \bar{y}_{..}) = \mu^{GS} + g_j^{GS}$$ \[20\]

and:

$$s_k = m(\bar{y}_{..k} = m\bar{y}_{..} + m(\bar{y}_{..k} - \bar{y}_{..})$$

$$= m\mu^{GS} + s_k^{GS}$$ \[21\]
In [19, 20, and 21], \( h_{GS}, \mu_{GS}, g_{j}^{GS} \) are the corresponding G-S solutions. Because G-S convergence was in one round, it is clear that \( J \) has not yet converged, as the \( g_{j} \) and \( s_{k} \) solutions differ.

The solutions for the second round of \( J \) would be obtained using [19], [20], and [21] jointly in [10], [11], and [12]. Thus:

\[
\begin{align*}
\bar{y}_{i}^{[2]} &= \sum_{j} g_{j}/G - \sum_{k} s_{k}/S \\
&= h_{i}^{GS} - \frac{1}{G} \left( \sum_{j} (\mu_{GS} + g_{j}^{GS}) \right) \\
&= \frac{1}{G} \left( \sum_{j} (m\mu_{GS} + s_{k}^{GS}) \right)/S \\
&= \frac{1}{G} \left( \sum_{j} (\mu_{GS} - m\mu_{GS}) \right) = h_{i}^{GS} - m\mu_{GS} \\
\end{align*}
\]

because the sums of the group and sire solutions in G-S are null, as shown before. Further:

\[
\begin{align*}
\bar{y}_{j}^{[2]} &= \sum_{i} \left( \mu_{GS} + h_{i}^{GS} \right)/H - \sum_{k} (m\mu_{GS} + s_{k}^{GS})/S \\
&= \frac{1}{H} \left( \sum_{i} (\mu_{GS} - m\mu_{GS}) \right) = g_{j}^{GS} - m\mu_{GS} \\
\end{align*}
\]

and:

\[
\begin{align*}
\bar{y}_{k}^{[2]} &= \sum_{i} \left( \mu_{GS} - m\mu_{GS} \right)/S \\
&= \frac{1}{S} \left( \sum_{i} (\mu_{GS} - m\mu_{GS}) \right) = s_{k}^{GS} - m\mu_{GS} \\
\end{align*}
\]

The term \( m\mu_{GS} \) appears in [22], [23], and [24], so \( J \) is diverging. The problem is caused by the "inability" of this method to handle unrestricted factors, i.e., factors that have the parameter \( \mu \) included. This can be remedied by redefining elements as \( g_{j}^{*} = g_{j} - g \) and \( s_{k} = s_{k} - s \), in which case \( J \) would converge in one round.

**Example**

The properties of G-S and J iterations are illustrated with the following example:

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>k</th>
<th>yijkl</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Assume that the starting solutions are null and that \( \lambda = 8 \). Using [7], [8], and [9] sequentially, G-S first round solutions are:

\[
\begin{align*}
h_{1} &= \frac{1}{4} \left( \sum_{ijkl} y_{ijkl} - g_{j} - s_{k} \right) \\
&= \frac{1}{4} \left( 17 + 10 + 12 + 7 \right) = 11.5 \\
\end{align*}
\]

\[
\begin{align*}
h_{2} &= \frac{1}{4} \left( \sum_{ijkl} y_{ijkl} - g_{j} - s_{k} \right) \\
&= \frac{1}{4} \left( 14 + 9 + 11 + 4 \right) = 9.5 \\
\end{align*}
\]

\[
\begin{align*}
g_{1} &= \frac{1}{4} \left( \sum_{ijkl} y_{ijkl} - h_{i} - s_{k} \right) \\
&= \frac{1}{4} \left( 17 - 11.5 \right) = \frac{1}{4} \left( 10 - 11.5 \right) + \frac{1}{4} \left( 14 - 9.5 \right) + \frac{1}{4} \left( 9 - 9.5 \right) = 2 \\
\end{align*}
\]

\[
\begin{align*}
g_{2} &= \frac{1}{4} \left( \sum_{ijkl} y_{ijkl} - h_{i} - s_{k} \right) \\
&= \frac{1}{4} \left( 12 - 11.5 \right) = \frac{1}{4} \left( 7 - 11.5 \right) + \frac{1}{4} \left( 11 - 9.5 \right) + \frac{1}{4} \left( 4 - 9.5 \right) = -2 \\
\end{align*}
\]

\[
\begin{align*}
s_{1} &= \frac{n_{s1}}{n_{s1} + \lambda} \left( \sum_{ijkl} y_{ijkl} - h_{i} - g_{j} \right) \\
&= \frac{4}{4 + 8} \left( 17 - 11.5 \right) + \frac{1}{4} \left( 12 - 11.5 \right) + \frac{1}{4} \left( 14 - 9.5 \right) + \frac{1}{4} \left( 11 - 9.5 \right) = 1 \\
\end{align*}
\]

\[
\begin{align*}
s_{2} &= \frac{n_{s2}}{n_{s2} + \lambda} \left( \sum_{ijkl} y_{ijkl} - h_{i} - g_{j} \right) \\
&= \frac{4}{4 + 8} \left( 10 - 11.5 \right) + \frac{1}{4} \left( 7 - 11.5 \right) + \frac{1}{4} \left( 9 - 9.5 \right) + \frac{1}{4} \left( 4 - 9.5 \right) = -1 \\
\end{align*}
\]

The next round of iteration gives the same solutions, which means that G-S converged in one round.
round. Note that \( g_1 + g_2 = 0 \) and \( s_1 + s_2 = 0 \). This illustrates that "sum to zero" constraints for \( g_j \) effects were imposed "automatically" by G-S. This is consistent with results of Van Vleck and Dwyer (7), who found that constraints did not increase the convergence rate of G-S.

In J iteration, the first round solutions are:

\[
\begin{align*}
h_1 &= \frac{17 + 10 + 12 + 7}{4} = 11.5 \\
h_2 &= \frac{14 + 9 + 11 + 4}{4} = 9.5 \\
g_1 &= \frac{17 + 10 + 14 + 9}{4} = 12.5 \\
g_2 &= \frac{12 + 7 + 11 + 4}{4} = 8.5 \\
s_1 &= \frac{4}{4 + 8} \left( \frac{17 + 12 + 14 + 11}{4} \right) = 4.5 \\
s_2 &= \frac{4 + 8}{4} \left( \frac{10 + 7 + 9 + 4}{4} \right) = 2.5
\end{align*}
\]

In the second round the solutions become:

\[
\begin{align*}
h_1 &= \frac{(17 - 12.5 - 4.5) + (10 - 12.5 - 2.5) + (12 - 8.5 - 4.5) + (7 - 8.5 - 2.5)}{4} = -2.5 \\
h_2 &= -4.5 \\
g_1 &= -1.5 \\
g_2 &= -5.5 \\
s_1 &= -2.5 \\
s_2 &= -4.5
\end{align*}
\]

In this example J is diverging. However, the first round solutions of both GS and J give identical values for: \( h_1, h_2, g_1 - g_2, s_1 - s_2 \). Jacobi iteration can converge in one round if the sums of both group and sire solutions are forced to zero:

\[
\begin{align*}
g_1^* &= g_1 - \sum g_j/G = 12.5 - (12.5 + 8.5)/2 = 2 \\
g_2^* &= g_2 - \sum g_j/G = 8.5 - (12.5 + 8.5)/2 = -2 \\
s_1^* &= s_1 - \sum s_k/S = 4.5 - (4.5 + 2.5)/2 = 1 \\
s_2^* &= s_2 - \sum s_k/S = 2.5 - (4.5 + 2.5)/2 = -1
\end{align*}
\]

It is not clear in unbalanced cases whether restrictions on fixed factors should be applied in subsequent rounds of iteration. The inefficiency of J in handling levels not expressed as deviations suggests that this should be done.

**EXTENSION TO AN ANIMAL MODEL**

Formulae [7], [8], and [9] can be extended to a model including a random factor having a general variance-covariance matrix, although none of the discussed method shall then converge in one round even if the data are balanced. Suppose in model [4] that the sire factor \( s \) is replaced by an animal factor \( a \). In this case, the inverse of an a matrix of additive relationships among animals is used to form the mixed model equations. Henderson (1) has given rules to form this inverse directly from a list of pedigreed animals, assuming no inbreeding. The coefficient matrix of the mixed model equations is amended as follows: 1) if none of the parents of the animal is known, add \( \lambda \) to the diagonal element of the equation for that animal; 3) if only one parent is known, add to the appropriate locations:

<table>
<thead>
<tr>
<th>Animal</th>
<th>Parent</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/3 ( \lambda )</td>
<td>-2/3 ( \lambda )</td>
</tr>
<tr>
<td>-2/3 ( \lambda )</td>
<td>1/3 ( \lambda )</td>
</tr>
</tbody>
</table>

3) if both parents are known, add:

<table>
<thead>
<tr>
<th>Animal</th>
<th>Parent 1</th>
<th>Parent 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 ( \lambda )</td>
<td>-( \lambda )</td>
<td>-( \lambda )</td>
</tr>
<tr>
<td>-( \lambda )</td>
<td>1/2 ( \lambda )</td>
<td>1/2 ( \lambda )</td>
</tr>
<tr>
<td>-( \lambda )</td>
<td>1/2 ( \lambda )</td>
<td>1/2 ( \lambda )</td>
</tr>
</tbody>
</table>

Let: \( a = \) effect of the animal under consideration, \( pr = \) effect of one parent, if the other is unknown, \( pr_1, pr_2 = \) effects of both parents, \( pg = \) effect of a progeny of a given animal, \( prg = \) effect of the other parent of the animal's progeny. Using these rules, the "contributions" of a given animal to its equation can be written as:
Type of contribution | Diagonal | LHS | RHS
--- | --- | --- | ---
1. Parents
Both parents known | $2\lambda$ | $2\lambda a - \lambda pr_1 - \lambda pr_2$ | ... 
One parent known | $\frac{4}{3} \lambda$ | $\frac{4}{3} \lambda a - \frac{2}{3} \lambda pr$ | ... 
Parents unknown | $\lambda$ | $\lambda a$ | ... 
2. Progeny
Other parent known | $\frac{1}{2} \lambda$ | $-\lambda pg + \frac{1}{2} \lambda a + \frac{1}{2} \lambda prg$ | ... 
Else | $\frac{1}{3} \lambda$ | $\frac{1}{2} \lambda pg + \frac{1}{3} \lambda a$ | ... 
3. Yield (if any) | 1 | $h_1 + g_j + a$ | $y_{ijkl}$

The diagonal ($d$) needed for G-S iteration is as in [2]:

$$d = n_a + \lambda \left( \frac{1}{2} n_1 + \frac{1}{3} n_2 \right) + \left\{ \begin{array}{c} 2\lambda \\
\frac{4}{3} \lambda \\
\lambda \
\end{array} \right. \tag{25}$$

where $n_a$ is the number of records of the animal, $n_1$ is the number of progeny of the animal with the other parent known, and $n_2$ is as $n_1$ but with the other parent unknown. The LHS is:

$$LHS = \sum_{l1}^{n_a} \left( (h_1 + g_j + a) + \lambda \left( \frac{1}{2} a - pg + \frac{1}{2} prg \right) + \lambda \left( \frac{1}{3} a - \frac{2}{3} pr \right) \right. \cdot \frac{1}{2} \left. \right. \right)$$

$\left. \right. \right.$

In [26], the sum is made over the records of the animal. Then $J$ or G-S iteration work with:

$$a = a + \left( \Sigma y_{ijkl} - LHS \right) / d \tag{27}$$

Example

Suppose iteration is already in progress so that the following current solutions of animals under evaluation are already available:

<table>
<thead>
<tr>
<th>Animal</th>
<th>Parent 1</th>
<th>Parent 2</th>
<th>Yield</th>
<th>Current solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>...</td>
<td>1.5</td>
</tr>
<tr>
<td>2</td>
<td>...</td>
<td>...</td>
<td>5</td>
<td>.8</td>
</tr>
<tr>
<td>3</td>
<td>...</td>
<td>4</td>
<td>7</td>
<td>.7</td>
</tr>
<tr>
<td>4</td>
<td>...</td>
<td>...</td>
<td>8</td>
<td>1.1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>4</td>
<td>...</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Consider the solution in the next round of iteration for animal 1. This animal has both parents known, and it has one progeny, with the other parent known. Then its diagonal element is:

$$d_1 = 0 + \lambda \left( \frac{1}{2} + 0 \right) + 2\lambda = 2.5\lambda$$

and:

$$LHS_1 = 0 + \lambda \left( \frac{1}{2} \times 1.5 - 1.2 + \frac{1}{2} \times 1.1 \right) + (0) + (2 \times 1.5 - .8 - .7) = 1.6\lambda$$

Thus:

\[ a_1 = 1.5 + \frac{(0 - 1.6\lambda)}{2.5\lambda} = .86 \]

**COMPUTING TECHNIQUES**

The averages [7], [8], and [9] can, in general, be computed in two ways: 1) accumulating appropriate sums and frequencies over the data file and calculating solutions at a later time, or 2) processing a file sorted for a particular factor; a solution for a single level becomes available immediately after processing the data pertaining to that level. The first way is easier to apply and is the only option available in J. The second way has smaller memory demands as accumulation storage for all levels of a single factor is not required. Further, it is the only option for G-S processing of factors with a relationship matrix. Here, the information should be available in duplicate form so that all information pertaining to a currently processed animal is accessible in a fast manner.

In general, in the G-S indirect approach one data file read per factor is needed in each round of iteration. Each read may apply either to the file in random order if appropriate accumulations are made or sorted for the processed factor; this is essential for a factor with a relationship matrix. For many factors, processing can be very slow. Schaeffer and Kennedy (4) proposed to solve only large factors by G-S and the smaller factors by inversion. In the J indirect approach, all sums and frequencies can be calculated during one read of the data file. To aid convergence rate, G-S and inversion methods may be mixed with J. Misztal (2) proposed to use G-S for a large factor with the data file sorted accordingly, while the other factors are solved by Jacobi. This still requires one file read per round of iteration, and the G-S part is able to remove the mean from the other factors efficiently.

Computing strategies differ in their memory requirements. In traditional approach or in indirect G-S with multiple sorted file, only one storage element per equation is necessary. In indirect G-S with an unsorted file read many times, storage for accumulating sums and frequencies for the largest factor must be added. Second-order J requires four storage elements per equation. Two of them contain old solutions and diagonals and are not needed until the end of each round. They can be stored on disk with little cost and effort.

Existing best linear unbiased prediction (BLUP) program can be modified to iterate on data rather than on the matrix of coefficients (3). Any BLUP program generates "contributions" to the system of equations. Let such "contributions" be \( a_{ij} \), \( r_{ik} \), where \( a_{ij} \) is the \( j \)th contribution to the coefficient matrix element \( a_{ij} \), and \( r_{ik} \) is the \( m \)th contribution to the RHS element \( r_i \). If the contributions are not stored but used to create a vector of diagonals \( d \) and a vector of accumulations \( c \) such that:

\[ d_i = \sum a_{ij} \]  
\[ c_i = \sum r_{ik} - \sum a_{ij} b_j^{[n]} \]

then iteration [3] can be rewritten as:

\[ b_{i}^{[n+1]} = b_{i}^{[n]} + \alpha(b_{i}^{[n]} - b_{i}^{[n-1]}) + c_i/d_i \]

**CONCLUSIONS**

Indirect approaches to computing solutions to mixed model equations are easier to understand and to program than direct "matrix" methods. However, the feasibility of their application depends on many factors. If efficient sorting is available, such as in mainframe computers, and the number of nonzero coefficients is small relative to the number of data points, the traditional approach can be considered. It will be particularly effective if a half-stored version of G-S is used, as suggested by S. P. Smith (1986, personal communication). If fast sorting is not available, e.g., in personal computers or supercomputers, indirect approaches should be used. These should also be employed with an animal model, where the number of records is usually smaller than the number of nonzero coefficients. With few factors in the model, second-order J mixed with G-S for one fixed factor should be most efficient. With many factors and extremely unbalanced data, G-S or SOR should be used for the most unbalanced factors. A factor with an additive relationship matrix should probably be always
computed by J or second-order J because it is much less laborious and not necessarily slower than G-S. For personal computers we suggest using unsorted data with Jacobi for the factor with an additive relationship matrix, and SOR for the other factors. This does not require sorting, it should converge for most layouts, and as many as 50,000 equations can be handled.

ACKNOWLEDGMENTS

This research was partly supported by Grant US-805-84 from BARD, The United States-Israel Binational Agricultural Research and Development Fund. The authors would like to express their gratitude to R. L. Fernando, who provided many useful comments.

REFERENCES