Conjugate gradient residual statics

Biondo Biondi

Stewart A. Levin

INTRODUCTION

Conjugate gradient methods are useful for solving large, generally sparse, minimization problems. For symmetric, positive-definite linear systems conjugate gradients is the optimal iterative technique when measured by a certain quadratic norm. Furthermore, as with Golub's QR method, conjugate gradients can solve least squares problems without squaring the condition number by forming normal equations.

One, now classic, least squares application in exploration geophysics is the residual statics problem. The paper of Wiggins, Larner and Wisecup (1977) clearly analyzes the algebraic structure of the problem and an iterative method (Gauss-Seidel) of solving the associated normal equations. In this report we will introduce two other iterative statics solution methods that are grounded in conjugate gradients. The first will simply apply conjugate gradients to solve the same normal equation system that had been solved by Wiggins, et al. by Gauss-Seidel. The second will not form the normal equations but instead will work directly with the original system from which the normal equations had been derived.

Our initial tests, surprisingly, do not show a striking improvement in quality or speed of convergence when conjugate gradients is applied to the problem. Using singular value decomposition on a model of modest size, we find that the diagonal preconditioning we employ in the conjugate gradient algorithm makes the eigenvalues of the normal equations very well behaved over about 80 percent of the spectrum. Since the Gauss-Seidel method is independent of diagonal scaling, this gives a plausible explanation for our results.

CLASSICAL FORMULAS

Following Wiggins, et al., let $x = \mathrm{col}(s, r, g, m)$. Let t_{ij} be the estimated static shift for shot i and receiver j. Then the basic residual statics formula

$$t_{ij} = s_i + r_j + g_k + m_k h_{ij}^2 \qquad , \tag{1}$$

where k is the CDP number and h_{ij} the source-receiver offset, describes a matrix-vector product Ax for producing time shifts from the statics model vector x. To apply conjugate gradients to residual statics we need to compute matrix-vector products $A^T t$ as well as Ax. It is straightforward to verify that this is given by:

$$\begin{aligned} s_i &= \sum_{i\ fixed} t_{ij} \\ r_j &= \sum_{j\ fixed} t_{ij} \\ g_k &= \sum_{k\ fixed} t_{ij} \\ m_k &= \sum_{k\ fixed} t_{ij}\, h_{ij}^2 \end{aligned} \qquad (2)$$

If we store i, j, k, h_{ij} in a table then the code for $y = y + Ax$ is simply:

```
      do 10 itrace=1,ntrace
      i=table(1,itrace)
      j=table(2,itrace)
      k=table(3,itrace)
      hij=table(4,itrace)
      y(itrace)=y(itrace)+s(i)+r(j)+g(k)+m(k)*hij**2
   10 continue
```

and the code for $x = x + A^T y$ is:

```
      do 20 itrace=1,ntrace
      i=table(1,itrace)
      j=table(2,itrace)
      k=table(3,itrace)
      hij=table(4,itrace)
      s(i)=s(i)+y(itrace)
      r(j)=r(j)+y(itrace)
      g(k)=g(k)+y(itrace)
      m(k)=m(k)+y(itrace)*hij**2
   20 continue
```

For the normal equations $A^T Ax$ needs to be formed. Using the above we may write this as

$$\begin{aligned} \hat{s}_i &= \sum_{i\ fixed} s_i + r_j + g_k + m_k\, h_{ij}^2 \\ \hat{r}_j &= \sum_{j\ fixed} s_i + r_j + g_k + m_k\, h_{ij}^2 \\ \hat{g}_k &= \sum_{k\ fixed} s_i + r_j + g_k + m_k\, h_{ij}^2 \\ \hat{m}_k &= \sum_{k\ fixed} h_{ij}^2 (s_i + r_j + g_k + m_k\, h_{ij}^2) \end{aligned} \qquad (3)$$

with code fragment

```
        call zero(snew,nshot)
        call zero(rnew,nrecv)
        call zero(gnew,ncdp)
        call zero(mnew,ncdp)

        do 30 itrace=1,ntrace
        i=table(1,itrace)
        j=table(2,itrace)
        k=table(3,itrace)
        hij=table(4,itrace)
        temp=s(i)+r(j)+g(k)+m(k)*hij**2
        snew(i)=snew(i)+temp
        rnew(j)=rnew(j)+temp
        gnew(k)=gnew(k)+temp
        mnew(k)=mnew(k)+temp*hij**2
30      continue
```

Besides the Gauss-Seidel iterative method, used by Wiggins et al., we will also work with the Jacobi method. To describe the Jacobi iteration, write $A^T A$ as $D + Q$ with $D = \mathrm{diag}(A^T A)$ and write $A^T t_{ij}$ as c. Then the Jacobi iteration is

$$Dx^{n+1} = c - Qx^n \tag{4}$$

with $x^0 = 0$. Since $A^T A$ depends only on the recording geometry, it is appropriate to precompute D^{-1}, i.e. the reciprocal of the diagonal elements. Then for the residual statics equations we may write the Jacobi iteration as

$$\begin{aligned}
\hat{s}_i &= ds_i^{-1} [\ cs_i - \sum_{i\ fixed} r_j + g_k + m_k h_{ij}^2\ ] \\
\hat{r}_j &= dr_j^{-1} [\ cr_j - \sum_{j\ fixed} s_i + g_k + m_k h_{ij}^2\ ] \\
\hat{g}_k &= dg_k^{-1} [\ cg_k - \sum_{k\ fixed} s_i + r_j + m_k h_{ij}^2\ ] \\
\hat{m}_k &= dm_k^{-1} [\ cm_k - \sum_{k\ fixed} h_{ij}^2 (s_i + r_j + g_k)\ ]
\end{aligned} \tag{5}$$

The Gauss-Seidel method splits the above into four separate iterations with the new $\hat{s}$ values being used to compute the new $\hat{r}$ values and so on. This has the advantage over the Jacobi iteration of using the most up-to-date information at each stage but the disadvantage of making the results depend on the order in which the variables are scanned. In matrix notation write $A^T A$ as $D + L + L^T$, where L is lower triangular; the Gauss-Seidel iteration is

$$(D + L)X^{n+1} = c - L^T X^n \quad . \tag{6}$$

For the purpose of applying the conjugate gradient method to the residual statics problem, the Jacobi method has one more decided advantage over the Gauss-Seidel method. The former, because D is symmetric and positive-definite, will preserve the key orthogonality properties of the conjugate gradient search directions. The latter does not and so the c-g optimality properties, based on orthogonality, are no longer valid.

Golub and Van Loan (1983) show that a symmetrized version of the Gauss-Seidel method (SSOR), wherein the order in which the variables are scanned is reversed from iteration, to iteration does produce a symmetric, positive-definite splitting and therefore may be accelerated by the same methods described below. We postpone discussion of this promising alternative to a future report.

ACCELERATED JACOBI

One conjugate gradient algorithm that directly compares with Gauss-Seidel incorporates the same diagonal matrix that appears in the Jacobi method. This algorithm applies conjugate gradients to the least-squares normal equations

$$A^T A x = A^T t \tag{7}$$

where x is as above and t is a vector of time shifts t_{ij} as determined by trace crosscorrelation. Schematically the algorithm is as follows (adapted, with apologies, from Golub and Van Loan, 1983):

```
$x = 0$; $r = A^T t$; $p = 0$; $\rho = 0$
for $k = 1,\ldots,$niter
    {
    $z = D^{-1} r$
    $\gamma = z^T r$
    $\beta = \gamma \rho$
    $p = z + \beta p$
    $w = A^T A p$
    $\alpha = \gamma / p^T w$
    $x = x + \alpha p$
    $r = r - \alpha w$
    $\rho = 1/\gamma$
    if $r \approx 0$ then break
    }
```

Because this algorithm specifically computes $A^T A$ times a vector we use it for illustration only. In practice one should use an alternative formulation which needs only multiplications by A or A^T but not by $A^T A$. One such algorithm is CGLS which take the form (adapted from Paige and Saunders, 1982):

```
$x = 0; p = A^T t; \gamma = p^T D^{-1} p$
for $k = 1, \ldots,$ niter
    {
    $s = D^{-1} p$
    $q = As$
    $\alpha = \gamma / q^T q$
    $x = x + \alpha p$
    $r = r - \alpha q$
    $s = A^T r$
    $\beta = \gamma$
    $\gamma = s^T D^{-1} s$
    $\beta = \gamma / \beta$
    $p = s + \beta p$
    if $s \approx 0$ then break;
    }
$x = D^{-1} x$
```

We emphasize that both conjugate-gradient algorithms will produce the same solution with exact arithmetic – the latter is preferred for its superior performance in the presence of rounding errors. All our examples are computed in double precision to remove the influence of such errors. As a further check we've also computed the solutions using the even more robust algorithm LSQR (Paige and Saunders, 1982). In all the examples we've run there has been no difference between the accelerated Jacobi outputs and the LSQR results.

As an aside, the SSOR acceleration mentioned in the introduction replaces D^{-1} with the new preconditioner $(D + L)^{-1} D (D + L^T)^{-1}$ which is implemented by cascading a forward and backward sweep of Gauss-Seidel and throwing out about half of the intermediate terms.

Our termination criterion for the iterations, both Gauss-Seidel and conjugate gradient, was to stop when the Euclidean norm of $A^T Ax - A^T t$ was reduced to less than 10^{-3} of the norm of $A^T t$. This is an overly stringent requirement. Accuracy of a few percent should be more than sufficient to produce an acceptable stack, the main goal of residual statics analysis.

SYNTHETICS

Our first synthetic is a small one with 4 shots and 6 traces per shot. The trace spacing and the shot pullup are both 1, as is the offset to the near trace. Figure 1 displays the g term as a function of iteration for Gauss-Seidel and accelerated Jacobi respectively. We can see that the two methods converge to answers that differ in both shape and magnitude. The difference involves two parts. The first is an endpoint contribution unresolvable by the residual statics model. This is completely lumped into the g term by the Gauss-Seidel iteration but split among all four terms by the accelerated Jacobi method. The second difference, associated with the small kink at midpoint 5, is a "long-wavelength" component of the solution that is poorly resolved by the residual statics data. The conjugate gradient algorithm picked up this component after 11 iterations as the residual norm dropped from half a percent to 10^{-5} in the last two iterations. The Gauss-Seidel method reached our specified tenth of a percent residual accuracy in only 8 iterations but the kink was pointing down and another 50 iterations were needed to bring it into agreement.

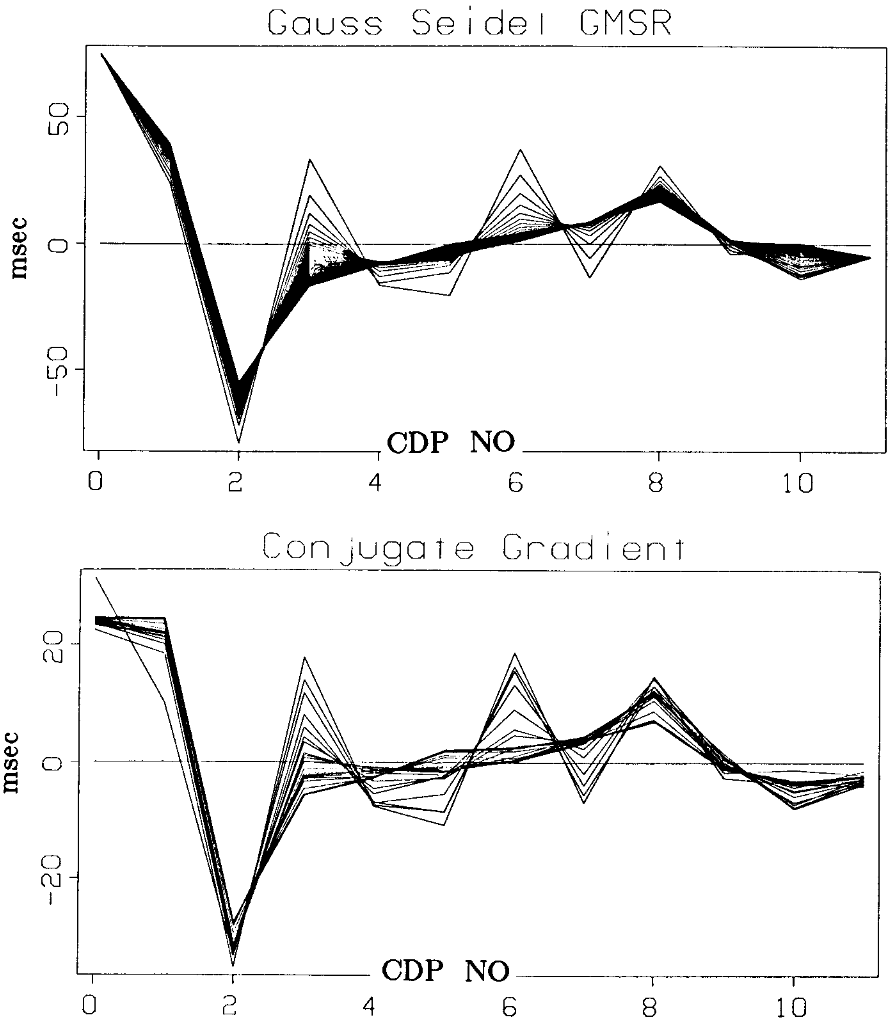

FIG. 1. The structure term of the solution for a small synthetic residual statics problem with 4 shots and 6 receivers; each curve is the result of one iteration. The upper plot displays the sequence produced by the Gauss-Seidel method while the lower is from conjugate gradients. We can see some differences attributable to the different convergence of the long wave-length components of the solution.

Our next synthetic has 64 shots and 24 traces per shot in a split spread geometry. The data were generated by adding random noise to a sinusoidal function with a two cable-length wavelength – a modest long wavelength problem. The magnitude of the noise was adjusted to make the signal to noise ratio about 10 to 1. In Figure 2 we show the shot statics converging to a solution. Both the Gauss-Seidel and conjugate gradient methods converged rapidly to solutions differing mostly by an unresolvable D.C. bias. When we changed the Gauss-Seidel order from $g \rightarrow m \rightarrow s \rightarrow r$ to $s \rightarrow r \rightarrow g \rightarrow m$, the number of iterations needed to reach an accurate solution decreased to two, a dramatic decrease to which we are unable to attach any significance.

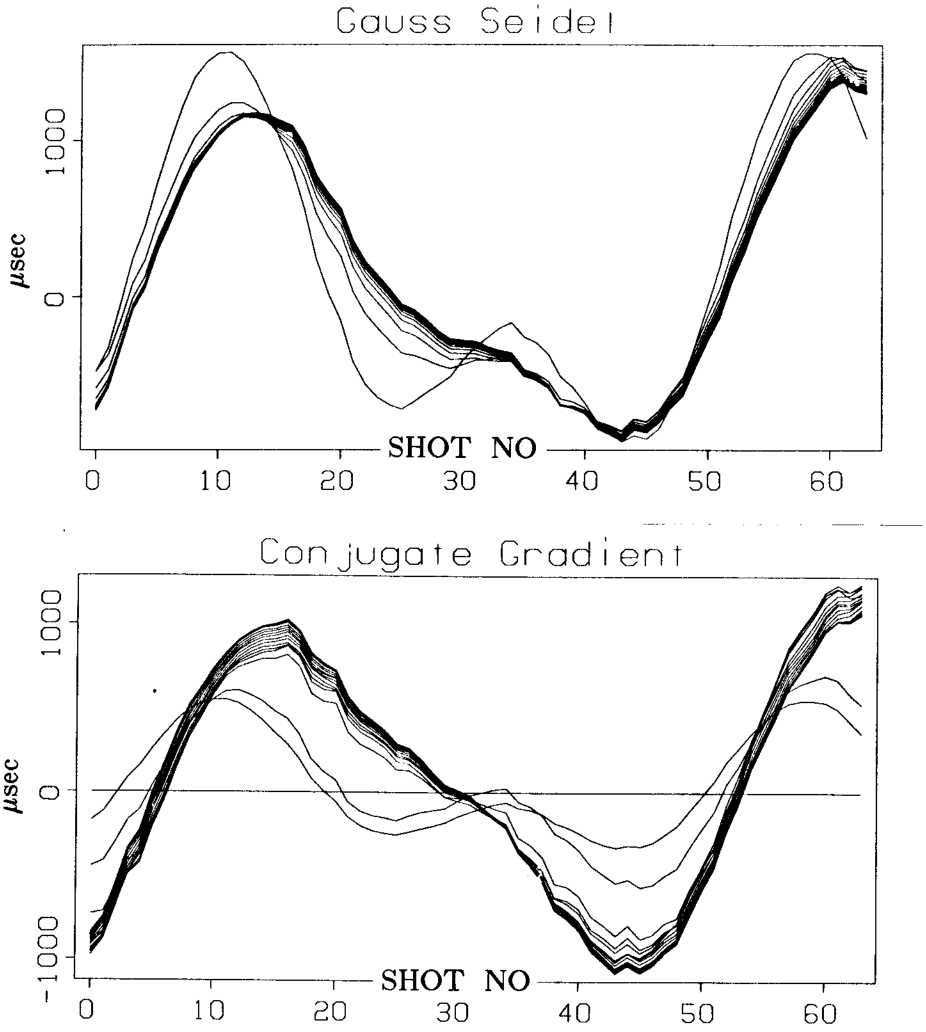

FIG. 2. The sequence of shot statics produced by Gauss-Seidel and conjugate gradient iterations for a synthetic problem with 64 shots and 24 traces per shot. The data were generated by adding random noise to a sinusoidal function with a two cable length-wavelength. The final results differ primarily by constant bias.

In Figure 3a we show an iterative solution for the m component when an end-on shooting geometry is used instead. This time the data were short period with a wavelength of half a cable length. Despite the ready resolvability of the short period solution, convergence was relatively slow and unstable near the ends of the lines where low folds prevail. This is accentuated in the RNMO term because in end-on shooting the midpoints gathers at one end of the line consist only of near offset traces and at the other end only of far offsets. This skewed offset-squared weighting of the RNMO explains the artifacts in the figure. By way of comparison, Figure 3b shows the m component when the same set of tij is assigned to a split-spread geometry. In this case the offset distributions are equally balanced at both ends of the lines and the iterations are better behaved.

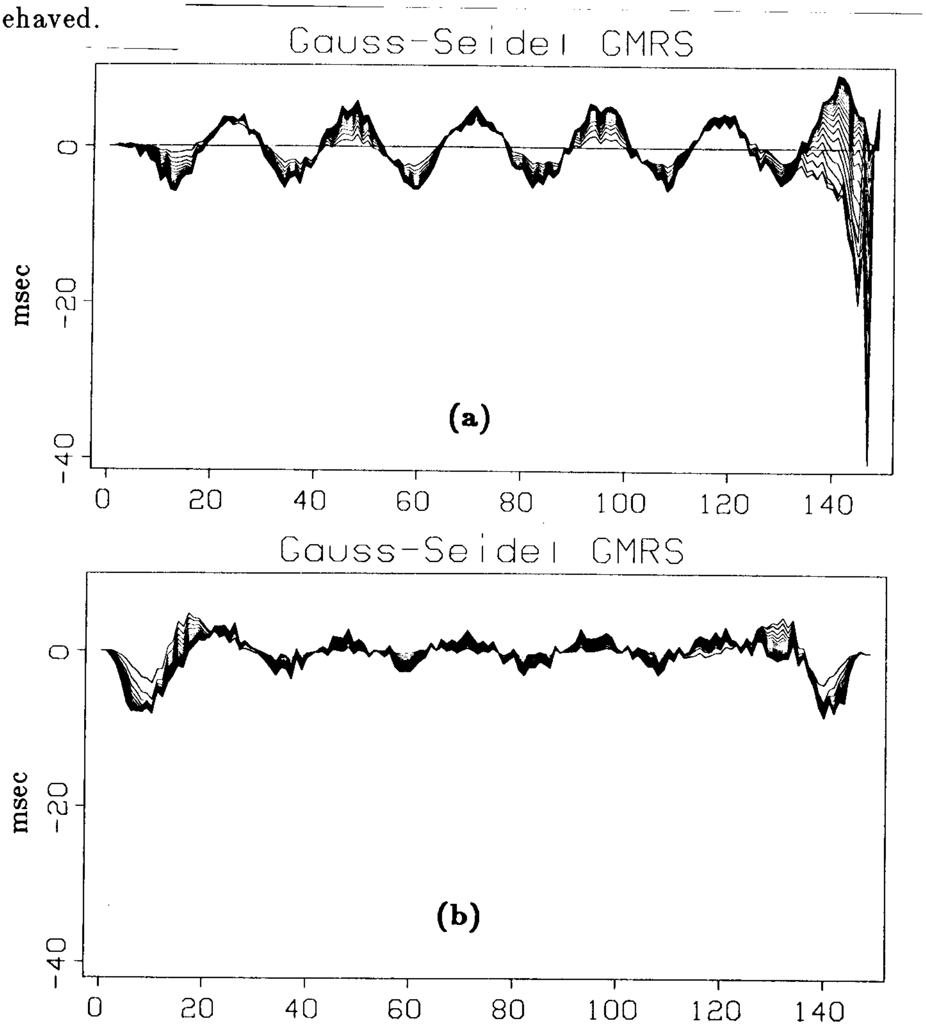

FIG. 3. a) The RNMO iterates for 64 shots and a 24 trace cable with end-on shooting geometry. The convergence was slow and unsteady at the end of the line. b) The same RNMO with split-spread geometry. The end points are better behaved and the same accuracy of fit was reached with fewer iterations.

Figures 3 actually raise two issues. The first is one of resolvability. As Wiggins, et al. discussed, the statistical uncertainty in a component eigenvector is inversely related to the corresponding eigenvalue of $A^T A$ which is fixed by the geometry of the seismic survey. Components with high variance are poorly resolvable despite any adjustments we make afterwards to our measurements. The second issue is that of scaling or preconditioning. Tapering the coefficients applied to first and last few model vector components is one form of preconditioning. Scaling each column to Euclidean norm 1 is a common preconditioning step. So is adding a small positive amount to each diagonal entry of $A^T A$. All of these schemes are designed to aid in finding a reasonable solution for our problem. The reliability of any given component in the final solution will not be changed.

SINGULAR VALUE DECOMPOSITION OF A

To figure out why we saw no remarkable improvement in accuracy and speed of convergence when we used conjugate gradients in our test cases above, we computed singular value decompositions (SVD) for the matrices A and $AD^{-1/2}$. We express the decomposition in the form

$$U^T \Lambda V \tag{8}$$

where Λ is a diagonal matrix containing the singular values, while U and V contain, respectively, the right and left singular vectors.

Figure 4 shows the singular values of the matrix A for 64 shots with 24 traces per shot. In this case A has 1536 rows and 451 columns. It took about 1000 CPU minutes to compute these using a canned program from the eispack library. In agreement with Wiggins, et al., we find there are 13 zero eigenvalues. At the other end of the spectrum we have 151 eigenvalues bunched around the value 200 followed by a rapid drop-off.

The matrix V corresponding to Figure 4 is shown in Figure 5. It is quite clear from this display that the large eigenvalues are associated with the residual moveout terms. This is a scaling problem – the coefficients of these terms in the matrix A are squared offsets for which we indeed used numbers on the order of 200. The coefficients of all the other terms are ones and zeros. The shot, receiver and structure components are significant only after the sharp decrease in the singular value curve. This means that the RNMO terms are the first to be determined in the iterative solution. This is not really what we would like; we'd rather determine the shots and receivers delays quickly because the structure and the RNMO terms may be refined in later post-stack processing and interpretation.

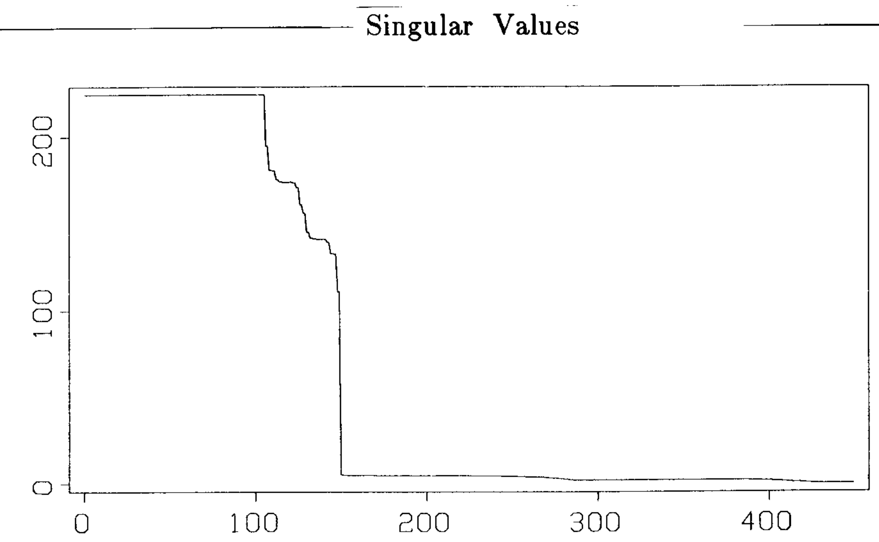

FIG. 4. Singular values for the residual statics matrix of the split-spread synthetic used in in Figure 2. The matrix is rank deficient by 13 so there are 13 singular values equal to zero.

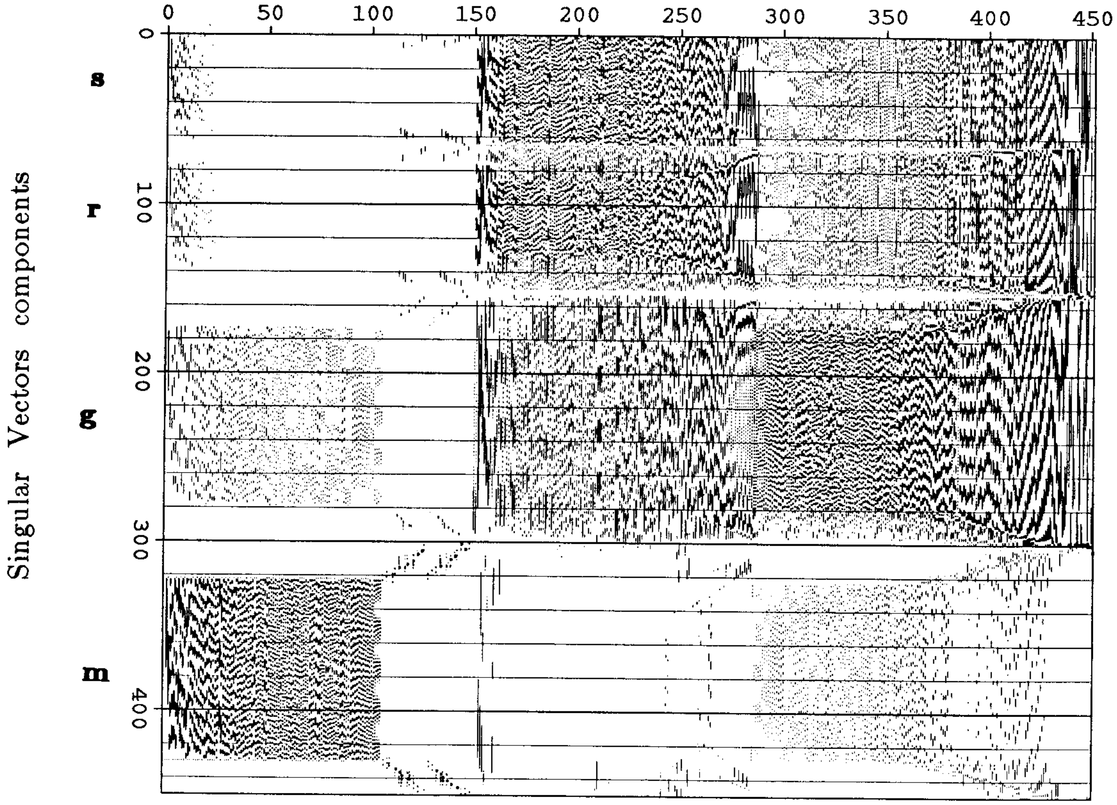

FIG. 5. Singular model vectors for the singular values of Figure 4. It is quite apparent that the large initial singular values are a result of the offset squared weighting of the m components.

The situation improves greatly when the diagonal, D, of $A^T A$ is used as a preconditioner. Figures 6 and 8 show the singular values for A and $AD^{-1/2}$ for a smaller synthetic with 12 shots and 10 traces per shot. The diagonal scaling improves the conditioning of the system – without scaling the singular values are reduced by a factor of 10 across the first third of the spectrum; with scaling we see only a factor of 4 across the first three quarters. This is about as well conditioned as we could reasonably hope with a matrix that's guaranteed to have 13 (about ten percent) zero eigenvalues. Any sensible iterative method (e.g. conjugate gradients and Gauss-Seidel) for solving this system will converge quickly. We further note that the Gauss-Seidel method iteration is independent of diagonal scaling in the following sense. Write

$$\begin{aligned} \tilde{D}^{-1} A^T A \tilde{D}^{-1} &= \tilde{D}^{-1} L \tilde{D}^{-1} + \tilde{D}^{-1} D \tilde{D}^{-1} + \tilde{D}^{-1} L^T \tilde{D}^{-1} \\ &= L_1 + D_1 + L_1^T \qquad . \end{aligned}$$

Then the Gauss-Seidel iteration matrix for the scaled system is

$$\begin{aligned} (D_1 + L_1)^{-1} L_1^T &= \tilde{D} (D + L)^{-1} \tilde{D} \tilde{D}^{-1} L^T \tilde{D}^{-1} \\ &= \tilde{D} \left[(D + L)^{-1} L^T \right] \tilde{D}^{-1} \end{aligned}$$

showing it is similar to the iteration matrix for the unscaled problem and hence has the same singular values.

If now we look at the corresponding V of the unscaled and scaled problems, shown in Figures 7 and 9, we see that the latter matrix is more balanced but the shots and receivers terms in the first singular vectors are still smaller than the structure and RNMO terms. This suggests we might still improve our choice of preconditioner to mix in more of the shot and receiver components in the early iterations. We'll examine in future work how SSOR preconditioning and various ad hoc smoothing constraints change the singular vectors.

SUMMARY

We now summarize what we have and haven't done in this study. We have described why it's worth looking at, and a method for, applying conjugate gradients to the residual statics equations. We've compared it with Gauss-Seidel on a few synthetics and found similar convergence and no great speedup using c-g. We've done SVD on one of the synthetics to discover that the residual statics matrix is quite well conditioned with the exception of a very few small singular values. This explains why conjugate gradients and Gauss-Seidel both converged quite rapidly. So far we've see no pressing reason for our sponsors to convert from Gauss-Seidel to c-g.

Now to what we haven't done. First, as Gene Golub has noted, conjugate gradients can be used to accelerate symmetric versions of Gauss-Seidel where one

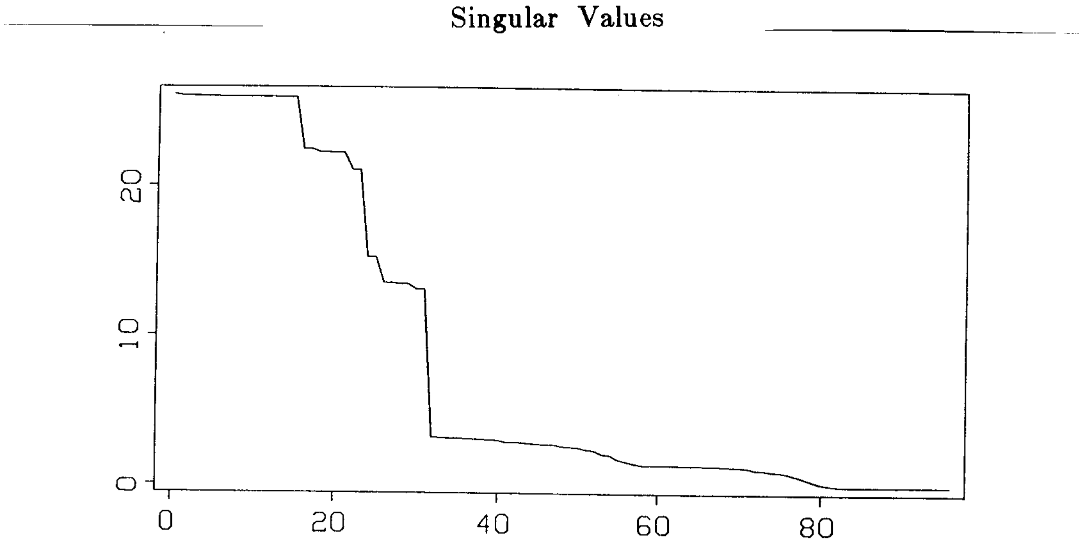

FIG. 6. Singular values for the coefficient matrix A with 12 shots and a 10 trace cable. The spectrum looks similar to that of the larger survey of Figure 4 but the scaling problems and sharp drop-off are not as severe.

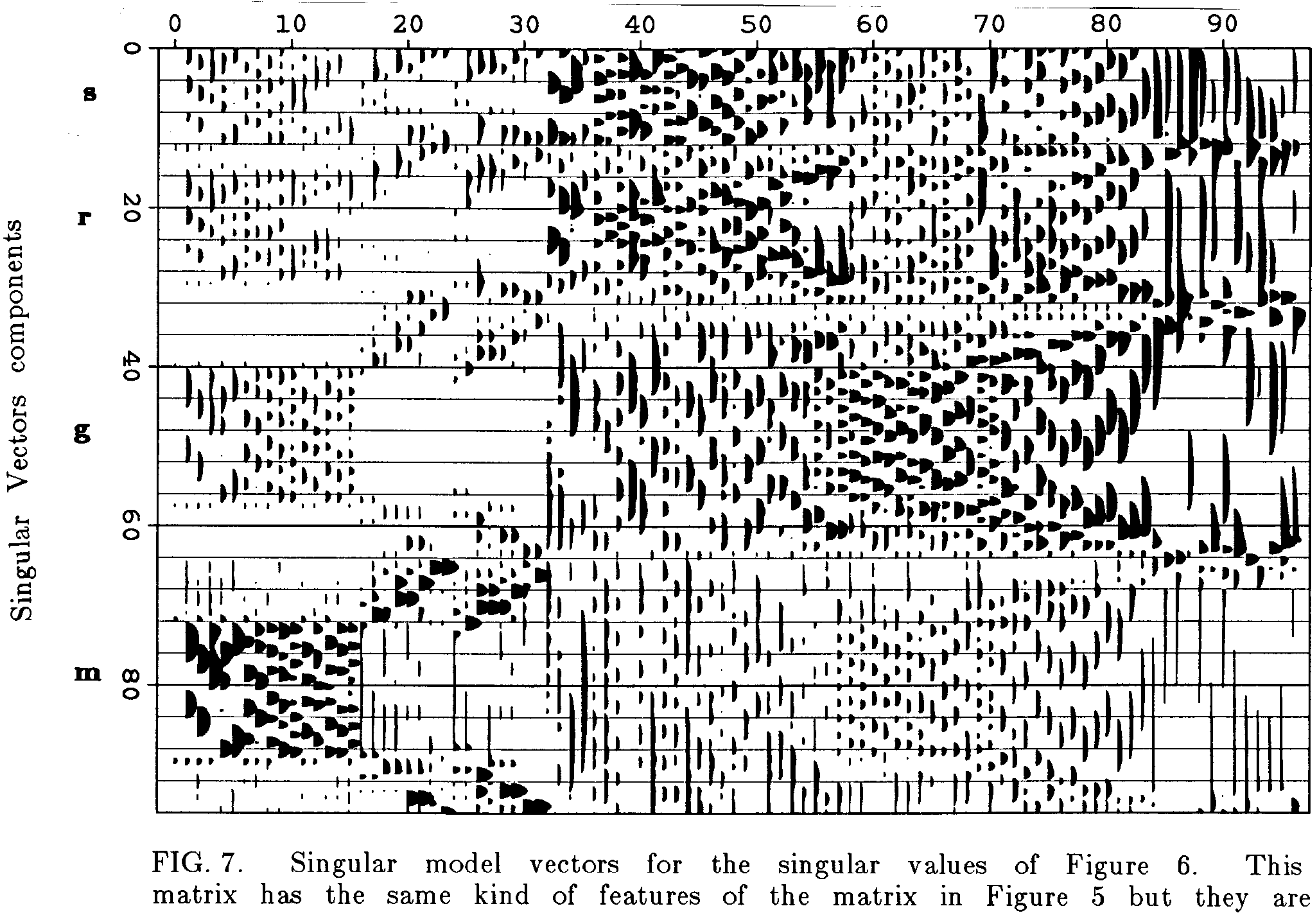

FIG. 7. Singular model vectors for the singular values of Figure 6. This matrix has the same kind of features of the matrix in Figure 5 but they are less pronounced.

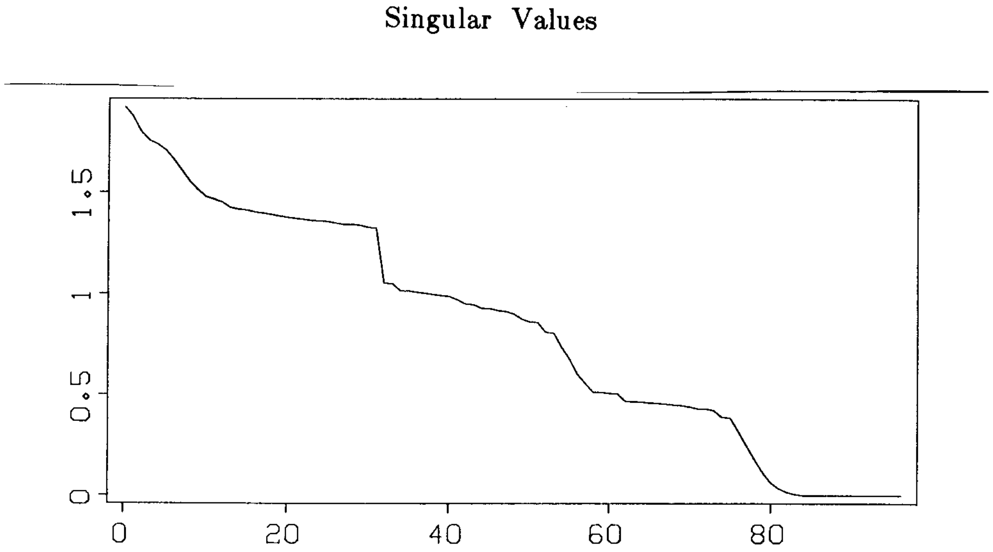

FIG. 8. Singular values after the the coefficient matrix A as in Figure 6 is preconditioned with the diagonal matrix $D^{-1/2}$. The spectrum now looks smoother and flatter, indicating the matrix is better conditioned.

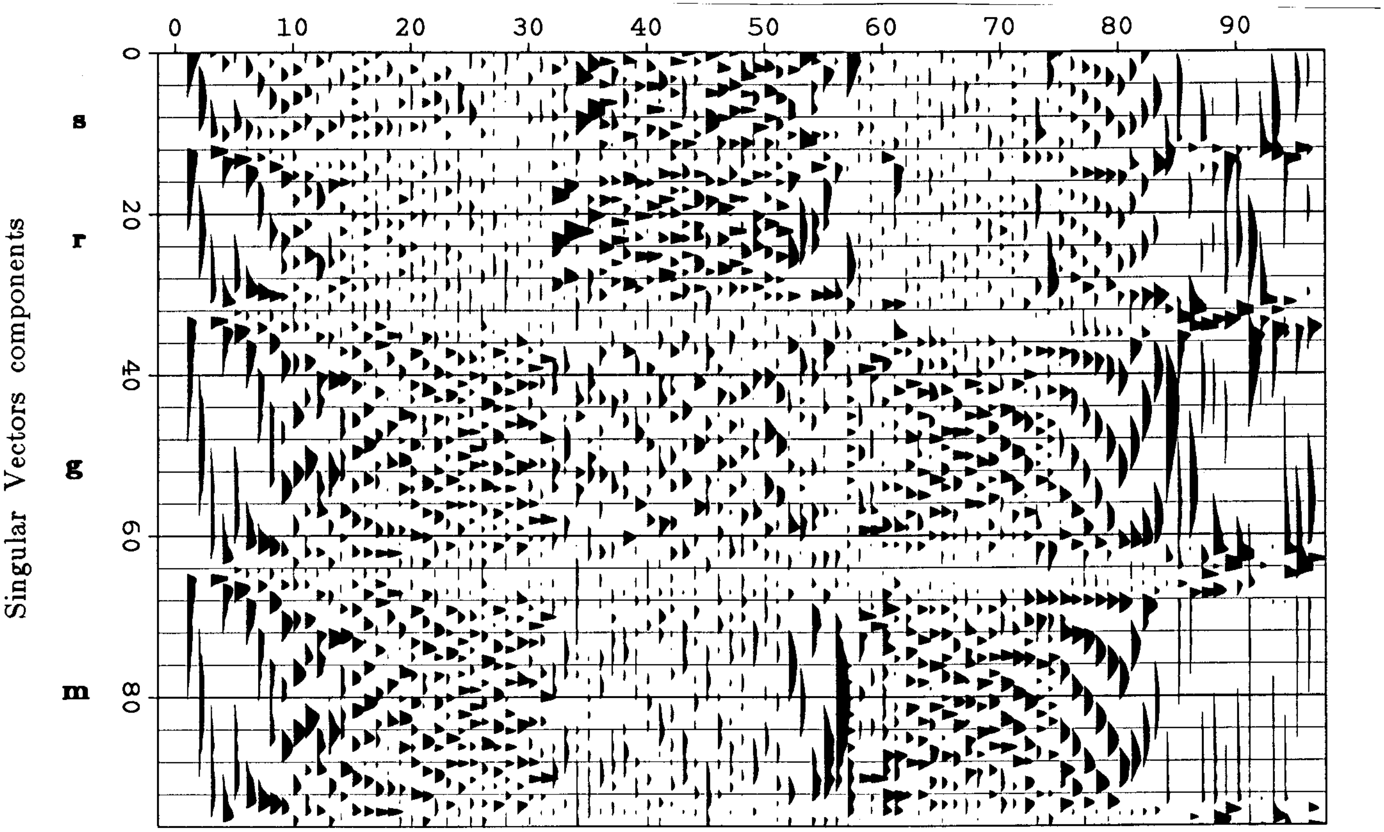

FIG. 9. Singular model vectors for the singular values of Figure 8. This matrix is more balanced than that shown in Figure 7 but the shots and the receivers components in the first singular vectors are still smaller than the structure and RNMO terms.

sweeps forward on the first iteration and backward on the second (e.g. $g\ m\ r\ s$ then $s\ r\ m\ g$). This acceleration we haven't done and is more directly comparable with Gauss-Seidel as done in industry practice. Second we haven't written any program to process field data and verify our conclusions about convergence and accuracy. This we consider an important step. Shuki Ronen found that in his experience with writing such a program, the bulk of the computation time is consumed in picking the t_{ij}'s. Improving the rate of convergence of the solution of the resulting residual statics equations will in that case produce only a small percentage overall savings. Lastly, we haven't seriously looked at the constraints routinely imposed in industry such as trend removal from the s and r terms and smoothing of the m terms and what they mean in terms of the residual statics matrix and its conditioning.

REFERENCES

Golub, G.H. and Van Loan, C.F., 1983, Matrix computations: Johns Hopkins University Press, Baltimore, 476 p.

Paige, C.C. and Saunders, M.A., 1982, LSQR: An algorithm for sparse linear equations ans sparse least squares: ACM Trans. Math. Soft., v 8 p 43-71.

Wiggins, R.A., Larner, K.L. and Wisecup, R.D., 1976, Residual statics as a general linear inverse problem: Geophysics v 41 p 922-938.