Comparison of Richardson's iteration with Chebyshev acceleration factors to conjugate gradient iteration

Christof Stork

ABSTRACT

Minimization of data variance may not be the best basis to judge iterative matrix inversion schemes. Instead, one may want to produce a *maximum likelihood* inversion and one may want to exercise control over the inversion. On this basis, Richardson's iteration with the application of Chebyshev acceleration factors has advantages over the conjugate gradient iteration for use in very large inverse problems that are common in geophysics.

INTRODUCTION

Richardson's method is a steepest decent iterative method with unit step size for locating the bottom of a quadratic, multidimensional surface that represents the error between data and model. The steepest decent method is generally taken to be the worst of many possible iterative methods for solving least-squares normal equations. The present, most popular method is probably the conjugate gradient method (Scales, 1987). However, the application of the Chebyshev acceleration factors (Olson, 1987) makes Richardson's method very competitive with the alternative techniques for applications with very large model spaces. Perhaps it should called a form of intelligent steepest descent.

The objective of conjugate gradient is to minimize data variance at each iteration step, which it does very well. To achieve this objective, the eigenvalue ranges inverted first are those that have greater data components. Larger eigenvalues may not be inverted before smaller ones, although a bias exists toward the larger eigenvalues.

Conjugate gradient iterations are generally continued until some stopping point is reached, which is frequently based on data variance reduction. When enough iterations can be performed to reach the stopping point, an alternative criterion can be used with a few additional iterations to produce a greater likelihood inversion (Aki and Richards, 1980) after a predefined number of iterations. Most greater likelihood inversions define the "best" inversion to be very accurate over some eigenvalue range with a smooth transition to the very small eigenvalues which are not inverted:

$$\lambda^G \approx \lambda^{-1} \quad \text{for } \lambda_{\min} < \lambda < \lambda_{\max} ,$$

and,

$$\lambda^G = 0 \quad \text{for } \lambda << \lambda_{\min} .$$

where:

λ^G = the generalized inversion of eigenvalue, λ.

In a maximum-likelihood inversion, such as the stochastic inverse ($\lambda^G = \frac{\lambda}{\lambda^2+\epsilon}$), larger eigenvalues are better inverted than the smaller ones.

Moreover, Richardson's iteration allows a user to control the inversion from his knowledge of the inverse application. For some cases such as the tomographic inversion used in Stork (1988a), this control is very desirable. However, Chebyshev acceleration factors require the maximum eigenvalue and the number of iterations be known before starting the iterations.

The behavior of these two iteration techniques is compared analytically and numerically. The analytic analysis introduces a different perspective than variance reduction for comparing of the two techniques.

The numerical comparison models the behavior of both methods for several different cases. Instead of constructing actual matrices, they are represented by their eigenvalue distribution.

ANALYTIC COMPARISON

Conjugate gradient is compared to Richardson's iteration with Chebyshev acceleration factors for a forward problem written as:

$$\Delta \mathbf{t} = \mathbf{L}\Delta \mathbf{s},$$

Conjugate gradient and Richardson's formulas are generally of the form:

$$\Delta \mathbf{s}^{(n+1)} = \Delta \mathbf{s}^{(n)} + \sigma_n \cdot \mathbf{g}_{(n)},$$

where:

$$\mathbf{g}_{(n)} = \mathbf{G}\Delta \mathbf{t}^{(n)} + \mu_n \cdot \mathbf{g}_{(n-1)}$$

σ_n and μ_n = scaling factors

$\mathbf{G}$= the gradient operator

$\Delta\mathbf{t}^{(n)}$= the residual data not explained by the previous iterations, $(\Delta\mathbf{t}^{(n)} = \Delta\mathbf{t}^{(0)} - \mathbf{L}\Delta\mathbf{s}^{(n)})$.

The gradient operator generally has the form:

$$\mathbf{G} = \mathbf{S}\mathbf{L}^T\mathbf{D},$$

where:

$\mathbf{D}$= a symmetric data weighting matrix

$\mathbf{L}^T$ = the transpose of the $\mathbf{L}$ matrix.

$\mathbf{S}$= a symmetric data weighting and preconditioning matrix. It is occasionally chosen to approximate $(\mathbf{L}^T\mathbf{D}\mathbf{L})^{-1}$ to speed the convergence of the iterations.

Conjugate gradient differs from Richardson's iteration in the scaling factors, σ and μ. For Richardson's iteration, $\mu_{(n)}$ equals 0.0 and $\sigma_{(n)}$ is the inverse of the square of the maximum eigenvalue ($\sigma_{(n)} = \frac{1}{\lambda^2_{\max}}$), while for the conjugate gradient method, $\mu_{(n)}$ and $\sigma_{(n)}$ are determined to minimize the square of the data mismatch: $\| \Delta\mathbf{t}^o - \mathbf{L}\Delta\mathbf{s}^{(n+1)} \|$. The application of Chebyshev acceleration factors to Richardson's iteration (Olson, 1987) uses scale factors of:

$$\sigma_n = \frac{2}{\cos\left(\frac{(2n+1)\pi}{2\cdot N}\right)\cdot(\lambda^2_{\max}-\lambda^2_{\min}) + (\lambda^2_{\max}+\lambda^2_{\min})}$$

$$\text{for: } n = 0, 1, \cdots N-1$$

The motivation of these acceleration factors will be presented later.

This generalized iteration formula can be manipulated into a more generic formula:

$$(\mathbf{S}^{-1/2}\Delta\mathbf{s}^{(n+1)}) = (\mathbf{S}^{-1/2}\Delta\mathbf{s}^{(n)}) + \sigma_n \cdot \mathbf{p}_{(n)}$$

and,

$$\mathbf{p}_{(n)} = (\mathbf{S}^{1/2}\mathbf{L}^T\mathbf{D}^{1/2})(\mathbf{D}^{1/2}\Delta\mathbf{t}^{(n)}) + \mu_{(n)}\mathbf{p}_{(n-1)}$$

By substituting $\mathbf{x} = \mathbf{S}^{-1/2}\Delta\mathbf{s}$, $\mathbf{b} = \mathbf{D}^{1/2}\Delta\mathbf{t}$, and $\mathbf{A} = \mathbf{D}^{1/2}\mathbf{L}\mathbf{S}^{1/2}$, we achieve a more compact form:

$$\mathbf{x}^{(n+1)} = \mu_n^{cg}\mathbf{x}^{(n)} + \sigma_n\,\mathbf{p}_{(n)}$$

$$\mathbf{p}_{(n)} = \mathbf{A}^T\mathbf{b}^{(n)} + \mu_{(n)}\mathbf{p}_{(n-1)},$$

which contains no preconditioning matrix, **S**, or data-weighting matrix, **D**. The square root of a symmetric matrix is defined as the square root of its eigenvalues with the eigenvectors kept intact.

This form converges to the least-squares solution of $\mathbf{b} = \mathbf{A}\mathbf{x}$ which is the original forward problem, $\Delta\mathbf{t} = \mathbf{L}\Delta\mathbf{s}$, with weights applied to the data and model space:

$$(\mathbf{D}^{1/2}\Delta\mathbf{t}) = (\mathbf{D}^{1/2}\mathbf{L}\mathbf{S}^{1/2})(\mathbf{S}^{-1/2}\Delta\mathbf{s}),$$

While the choice of the preconditioning matrix, **S**, may speed convergence, it may produce very undesirable weights. Strong caution is urged in choosing the preconditioning. For instance, a common choice for the preconditioning is too boost the high wavenumbers of an inversion since they are inverted slower. However, this preconditioning encourages high wavenumber components in the result, which is generally not desirable.

The generic conjugate gradient equation involves only the matrix $\mathbf{A}^T$, which when multiplied with **b** produces the gradient of steepest decent down the quadratic error function, $E = (\mathbf{b}^{(0)} - \mathbf{A}\mathbf{x})^2$. The name "conjugate gradient" results from the method using a linear combination of the present gradient, $(\mathbf{A}^T\mathbf{b}^{(n)})$, and last gradient, $(\mathbf{p}^{(n)})$. Each new gradient is perpendicular to and properly scaled in relation to all the others. Once enough vectors have been produced to span the entire resolved space, they eliminate all possible data variance. Steepest descent iteration simply marches down the contours of the error function.

The methods can be compared using the variance perspective for the two dimensional case using Figure 1 borrowed from Figure 2.10 of John Toldi's thesis

(1985). Conjugate gradient will take only two steps to reach the bottom of the two dimensional objective function while steepest-descent will take an infinite number of iterations as it zig-zags down. From this perspective, conjugate gradient appears to have clear advantages.

An alternative perspective for analyzing the iteration techniques is the inversion of the eigenvalues, independent of the data components. The effect of the iterations on the eigenvalues is analyzed with the following decomposition performed with the insight of Comer and Clayton (1986) and Ivansson (1983). It is this perspective from which the Chebyshev scaling factors are chosen for Ricardson's iteration.

Continuing from the equations of before,

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \sigma_n \cdot \mathbf{g}_{(n)}$$

$$\begin{aligned}\mathbf{g}_{(n)} &= \mathbf{A}^T \left(\mathbf{b}^{(0)} - \mathbf{A}\mathbf{x}^{(n)}\right) + \mu_n \cdot \mathbf{g}_{(n-1)} \\ &= \mathbf{A}^T \mathbf{b}^{(0)} - \mathbf{A}^T \mathbf{A}\mathbf{x}^{(n)} + \mu_n \cdot \mathbf{g}_{(n-1)}\end{aligned}$$

Combining these equations, produces:

$$\mathbf{x}^{(n+1)} = \left(\mathbf{I} - \sigma_n \cdot \mathbf{A}^T \mathbf{A}\right)\mathbf{x}^{(n)} + \sigma_n \cdot \mathbf{A}^T \mathbf{b}^{(0)} + \sigma_n \cdot \mu_n \cdot \mathbf{g}_{(n-1)}$$

Careful consideration convinces us that recursively substituting in for $\mathbf{x}$ and $\mathbf{g}$ until we reach $\mathbf{x}^{(0)}$ and $\mathbf{g}_{(0)}$, which are defined to be zero, will produce a result of the following form:

$$\mathbf{x}^{(n)} = \left(\sum_{i=0}^{n} \beta_i \cdot (\mathbf{A}^T \mathbf{A})^i\right) \cdot \mathbf{A}^T \mathbf{b}^{(0)}$$

where the coefficients β_i will be some function of the σ's and μ's. An unsuccessful attempt was made in appendix B to find an explicit expression for the β's.

By substituting in the singular value decomposition for $\mathbf{A}$, $\mathbf{A} = \mathbf{U}\mathbf{A}\mathbf{V}^T$, this form is rewritten:

$$\mathbf{x}^{(n)} = \mathbf{V}\left[\sum_{i=0}^{n} \beta_i \cdot \Sigma^i\right]\Sigma \mathbf{U}^T \mathbf{b}^{(0)}$$

Thus, the generalized inversion achieved for each eigenvalue after n iterations can be represented as a polynomial:

$$\lambda^G = \sum_{i=0}^{n} \beta_i \cdot \lambda^i \cdot \lambda$$

We can choose the coefficients, β_i, to best match the desired eigenvalue inversion, such as the stochastic maximum likelihood inversion, $\lambda^G = \frac{\lambda}{\lambda^2 + \epsilon}$, for example. How accurately we match our desired function is dependent on the degree of our polynomial, which is the number of iterations we perform.

If we could determine the scaling factors necessary to produce these coefficients, we would have an intelligent form of steepest decent. Unfortunately, converting these coefficients, β_i, back into scaling factors used in Richarson's iteration is not an easy task. However, Olson (1987) is able to determine an effective method for defining the Richardson's scaling factors based on Chebyshev polynomials. The development is presented in appendix C. The development also suggests how any n'th order polynomial can be represented only with Richardson's iteration.

The Chebyshev scaling factors will match the function $\lambda \cdot \lambda^G = 1.0$ with approximately even error over a specified eigenvalue range. Figure 2 is plot of the function $\lambda \cdot \lambda^G$ for two different sets of Chebyshev scale factors, chosen with different minimum eigenvalues. The height of the peaks and troughs for each set of iterations are identical over the eigenvalue range . A trade off exists between accuracy of the inversion and eigenvalue range over which the inversion is performed. These scale factors provide a smooth transition to $\lambda^G = 0$ for very small λ.

Since any polynomial,

$$\lambda^G = \sum_{i=0}^{n} \beta_i' \cdot \lambda^i \cdot \lambda$$

can be represented with Richardson's iteration, any conjugate gradient inversion can be identically reproduced with Richardson's iteration. This relationship may not be directly apparent, but can be seen that only one scaling factor per iteration is needed to have enough degrees of freedom after n iterations to define an n'th order polynomial. Although we cannot determine the exact relationship between the scaling factors of Richardson's iteration with those of conjugate gradient, they will be different since they are chosen to satisfy two different types of criterion.

The conjugate gradient method minimizes data variance at each step; the Chebyshev acceleration factors are the optimal scale factors based on three conditions:

Figure 1: Figure 2.10 borrowed from John Toldi's thesis. The contours represent the variance of a two dimensional inverse problem. Steepest decent (solid line) takes an infinite number of iterations to reach the bottom as it zig-zags down. Conjugate gradient (dashed line) reaches the bottom in two steps. Additional dimensionals complicate this model.

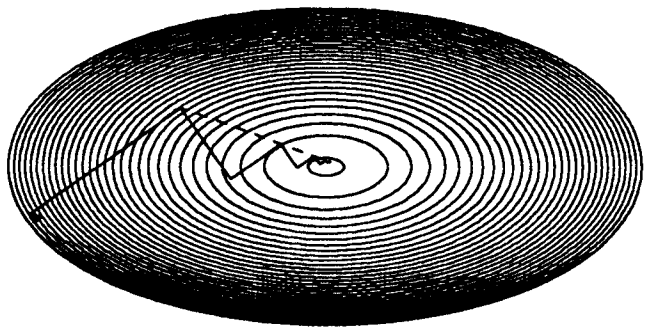
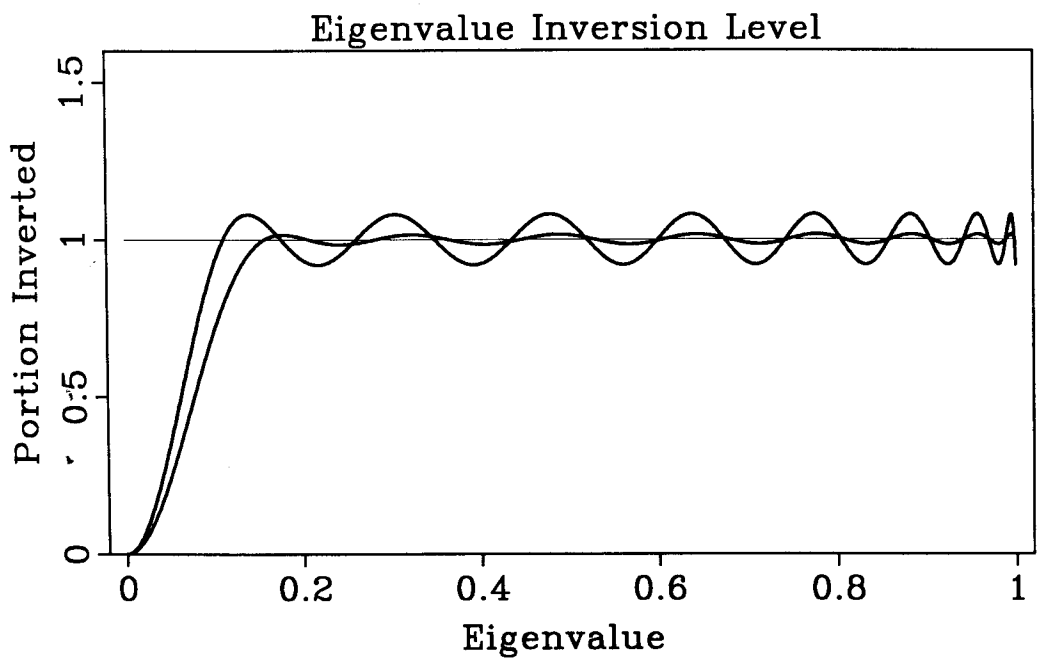


Figure 2: The inversion level ($\lambda \cdot \lambda^G$) for two sets of 16 iterations of Richadson's iteration with Chebyshev scaling factors. The maximum error of all the bumps for one set of inversion is even. A trade-off exists between the size of the bumps and the eigenvalue range inverted.



1) a predetermined number of iterations, **2)** an eigenvalue range the inversion is desired over, and **3)** the minimization of the maximum eigenvalue error over the specified range, i.e., $\min(\max \mid 1.0 - \lambda \cdot \lambda^G \mid)$ where, $\lambda^G \approx \lambda^{-1}$ for $\lambda_{min} < \lambda < \lambda_{max}$.

NUMERICAL COMPARISON

Conjugate gradient iteration is compared with Chebyshev iteration for several sample matrix inversions. Instead of actually creating sample matrices and performing the iterative inversion explicitly, the inversion procedure can be completely represented by considering only the eigenvalue distribution of the matrix, **A**, and the distribution of the energy in the data over these eigenvalues. The inversion procedure is represented by the inversion of the individual eigenvalues as described in Stork, (1988b).

The behavior of the inversions is presented according to two characteristics: variance reduction, and inversion of the eigenvalues. These two characteristics represent the different perspectives the scaling factors of the two iteration methods are defined in. The parameters of conjugate gradient are chosen based on the variance reduction, while those of Richardson's iteration are chosen based only on the inversion of the eigenvalues.

These two basises for comparison are, of course, closely related. The variance reduction is a function of the inversion of the eigenvalues and the distribution of the data energy over the eigenvalues range. In the conventional form, variance is:

$$variance = \mid \mid \mathbf{b}^{(0)} - \mathbf{A}(\mathbf{A}^G \mathbf{b}^{(0)}) \mid \mid$$

To treat the distribution of data energy as a continuum over the eigenvalue range, it is written as a function: $E(\lambda)$, where the distribution of eigenvalues is included in the function. The development of this function is presented in appendix A. The variance can be written as:

$$variance = \int_{\lambda_{min}}^{\lambda_{max}} E(\lambda) \cdot (1.0 - \lambda \cdot \lambda^G)^2 \cdot d\lambda$$

where :

λ^G represents the generalized inversion of λ.

$E(\lambda)$ is the distribution of the data energy over the eigenvalue range.

The behavior of conjugate gradient and Richardson's iteration with Chebyshev acceleration factors is compared for various different data energy distributions. The first comparison will be for a constant data energy distribution, $E(\lambda) = 1$.

Figures 3a and b show the eigenvalue inversion level after each iteration of conjugate gradient and Chebyshev iterations. The eigenvalue inversion level is defined as: $\lambda \cdot \lambda^G$, where λ^G represents the generalized inversion achieved after the iterations. The method for its computation is presented in Stork (1988b).

The function $\lambda \cdot \lambda^G$ should equal 1.0 for the large eigenvalues and 0.0 for the very small ones. The thick black line in Figures 3a and 3b is the result after 16 iterations. The conjugate gradient method is stable after each iteration, as would be expected. The Chebyshev method, however, gyrates wildly until the last iteration.

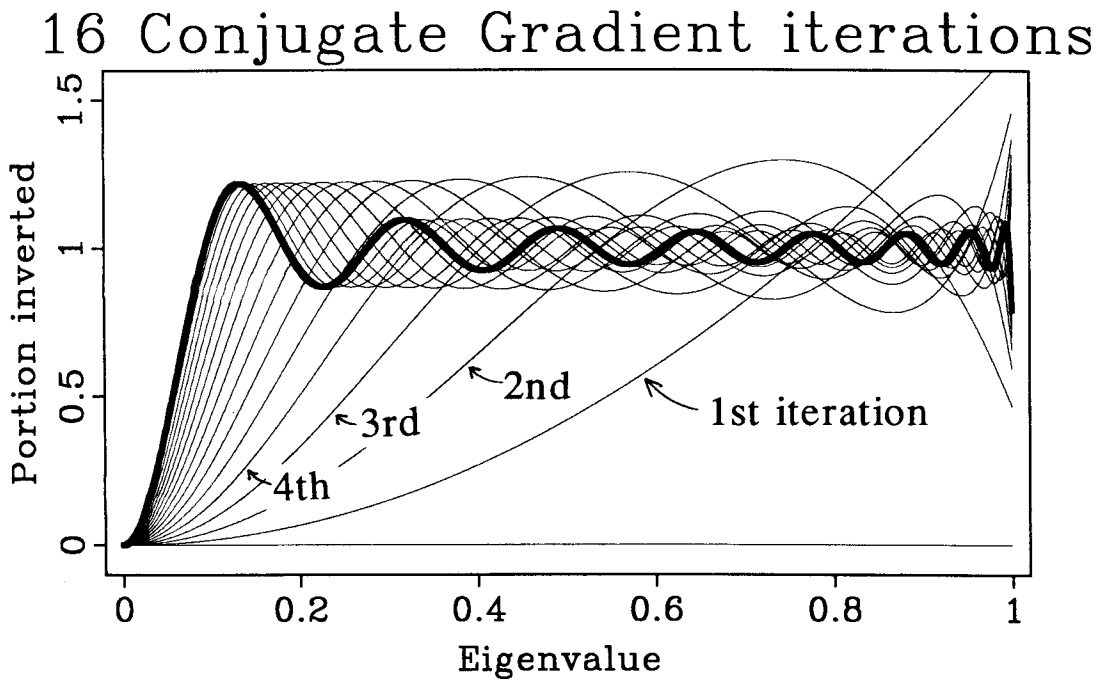
The result after 16 iterations for both techniques are plotted together in Figure 4b. The conjugate gradient method has inverted to smaller eigenvalue, but to less accuracy over the eigenvalue range. The peaks and troughs of the Chebyshev iterations are even over the eigenvalue range, while they are greater at the low eigenvalues for the conjugate gradient iterations. Note also that conjugate gradient has not inverted the eigenvalue of 1.0 very accurately.

Comparison of the variance reduction is shown in Figure 4c. The residual variance of the conjugate gradient iterations is lower than for Chebyshev iterations after 16 iterations, although they are quite close. They gyrations of the Chebyshev iterations is apparent, but it finally produces a reasonable result after the set number of iterations.

The eigenvalue range inverted with Chebyshev iterations is adjusted to invert to smaller eigenvalue similar to that of the previous conjugate gradient iterations. The result is shown in Figure 5b. Conjugate gradient has inverted the larger eigenvalues slightly more accurately than Chebyshev, but not the lower eigenvalues. The variance reduction, plotted in Figure 5c, shows the two produce nearly the same result after 16 iteration, but the path to the result is very different. Conjugate gradient is stable for all iterations; the Chebyshev method is stable only after the last iteration.

The behavior of conjugate gradient iteration is affected by the the eigenvalue distribution of the problem and of the data content. This effect is seen with the use of Figure 6a, b, and c. The data energy distribution, $E(\lambda)$, consists of four bands seen in Figure 6a. The conjugate gradient iteration has accurately inverted the eigenvalues

Figure 3: Inversion level ($\lambda \cdot \lambda^G$) after each step of iterations for **a)** conjugate gradient iteration and **b)** Chebyshev iteration. Thick line is the result after 16 iterations. Conjugate gradient produces a stable result after every iteration while Chebyshev does not. Chebyshev produces a reasonable result only after the last iteration.



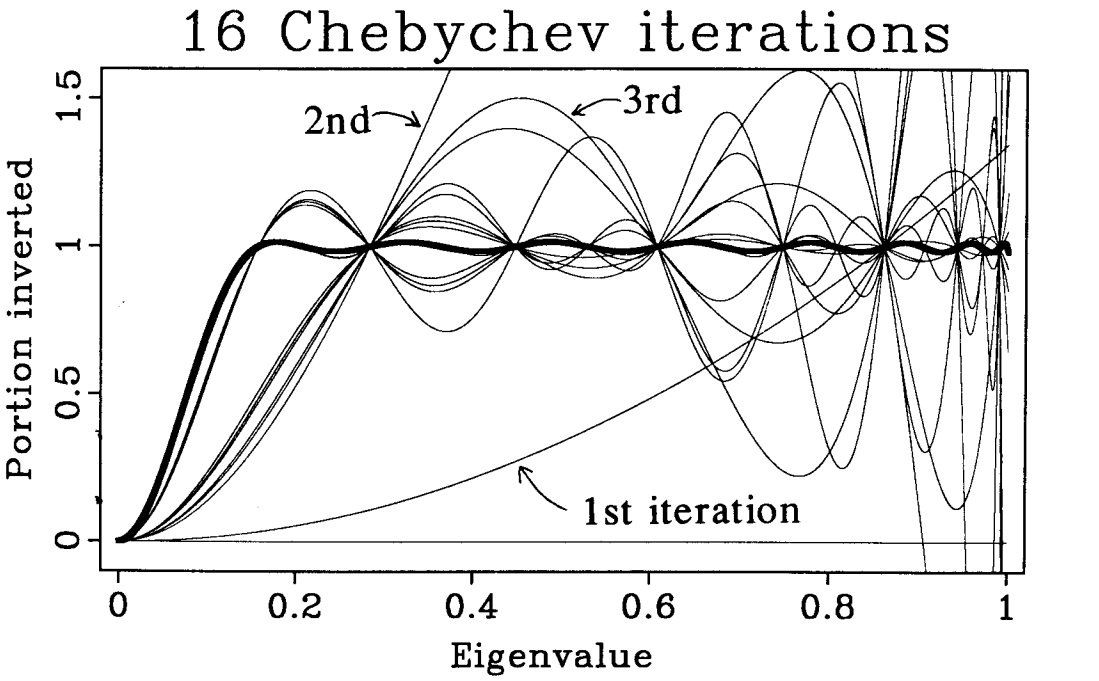


Figure 4: **a)** Even data energy distribution, $E(\lambda)$, used to compare conjugate gradient and Chebyshev iteration. **b)** Inversion level $(\lambda \cdot \lambda^G)$ for conjugate gradient and Chebyshev iteration. Conjugate gradient has inverted to smaller eigenvalue but Chebyshev has inverted the larger eigenvalues more accurately. **c)** Residual variance after each iteration. Conjugate gradient follows a stable path while Chebyshev does not, producing a reasonable result after only the last iteration. Conjugate gradient has done a better job after the 16 iterations.

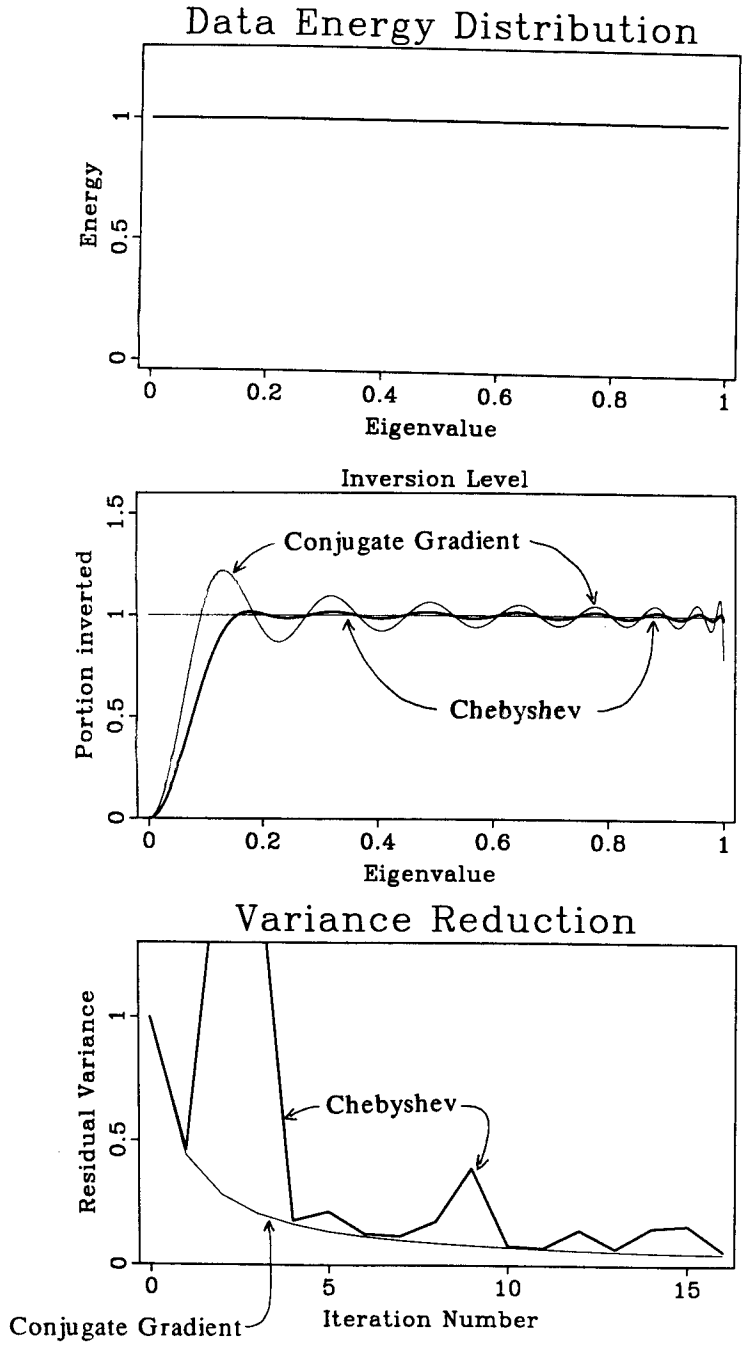
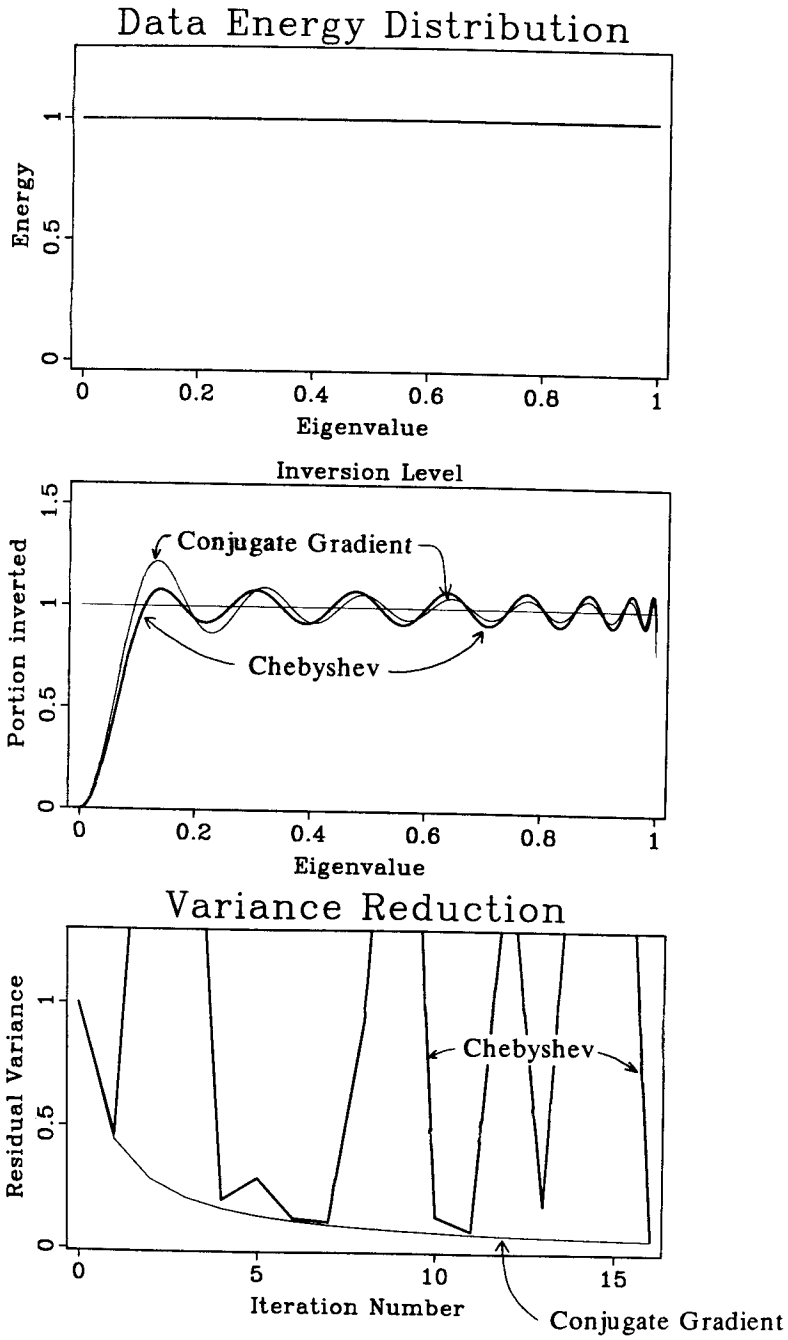


Figure 5: **a)** Even data energy distribution, $E(\lambda)$, used again to compare conjugate gradient and Chebyshev iteration. **b)** Inversion level for conjugate gradient and Chebyshev iteration. This plot is similar to that of Figure 4b except that the scaling factors of Chebyshev iteration were chosen to invert to smaller eigenvalue. Conjugate gradient has smaller bumps at the larger eigenvalues, but larger bumps at the smaller eigenvalues. It has also inverted to slightly smaller eigenvalue. better **c)** Residual variance after each iteration. Chebyshev iterations are now very unstable until the last iteration, when it's variance reduction is nearly the same as that for conjugate gradient.



only within these bands, seen in Figure 6b. The positions of these bands is also plotted in figure 6b. Since there are no data components in the region between these bands, their poor inversion has no impact on the result. The Chebyshev scaling factors are chosen irrespective of the data energy distribution and invert the regions with no data energy as well as those with data. Since conjugate gradient was able to "conserve" its iterations, it was able to partially invert even the band at very low eigenvalue.

Analysis of the variance reduction shows that conjugate gradient has done a much better job than the Chebyshev iterations. Since the Chebyshev iteration inverted the eigenvalues better in the three bands with larger eigenvalues, the greater variance reduction of conjugate gradient comes almost entirely from the better inversion of the band at smallest eigenvalue.

The final energy distribution function considered, shown in Figure 7a, has most of the data energy at small eigenvalue. The conjugate gradient iterations do not invert the very large eigenvalues well. Otherwise, Chebyshev iteration was very similar, except that it didn't invert to quite as small of an eigenvalue. The plot of the residual data variance shows the end result to be similar, although conjugate gradient is slightly lower.

DISCUSSION

In all the numerical examples above, conjugate gradient has been more successful at reducing data variance. However, the inversion result may not be more desirable. In Figure 6b, conjugate gradient has inverted the three bands at larger eigenvalue less well than Chebyshev, but has inverted the band at very low eigenvalue, below 0.05. The signal to noise level may not justify the inversion of such a small eigenvalue, but conjugate gradient does not give us the option to avoid inversion of these eigenvalues. Figure 7b shows that conjugate gradient has "sacrificed" the inversion of the larger eigenvalues for the lower ones. While they contain only small data components, they are the best resolved part of the inversion problem. If we are seeking the most accurate solution, we will still want to accurately invert these data components even if it means performing a few additional iterations.

Inverse theory tell us that a maximum likelihood inversion requires larger eigenvalues be inverted better than smaller ones, regardless of data energy distribution. If the computer power is available, which it generally is even for very large problems such as tomographic inversion, it may be advantageous to perform the extra iterations necessary to perform a maximum likelihood inversion. Data variance of the Chebyshev method was not significantly poorer than that for the conjugate gradient method which suggests only a few additional iterations would be needed to achieve a similar variance reduction but with a maximum likelihood inversion result.

Figure 6: **a)** Data energy distribution with four bands. **b)** Inversion level for conjugate gradient and Chebyshev iteration. Conjugate gradient has not bothered to invert the eigenvalues with no data energy, while Chebyshev iteration is not effected by the data energy distribution. Conjugate gradient has inverted some of the band at very small eigenvalue while Chebyshev has not. **c)** Conjugate gradient has produced a much lower variance than Chebyshev after 16 iterations. Since Chebyshev has better inverted the eigenvalue in the three upper bands, the lower variance for conjugate gradient must all result from its better inversion of the band at smallest eigenvalue.

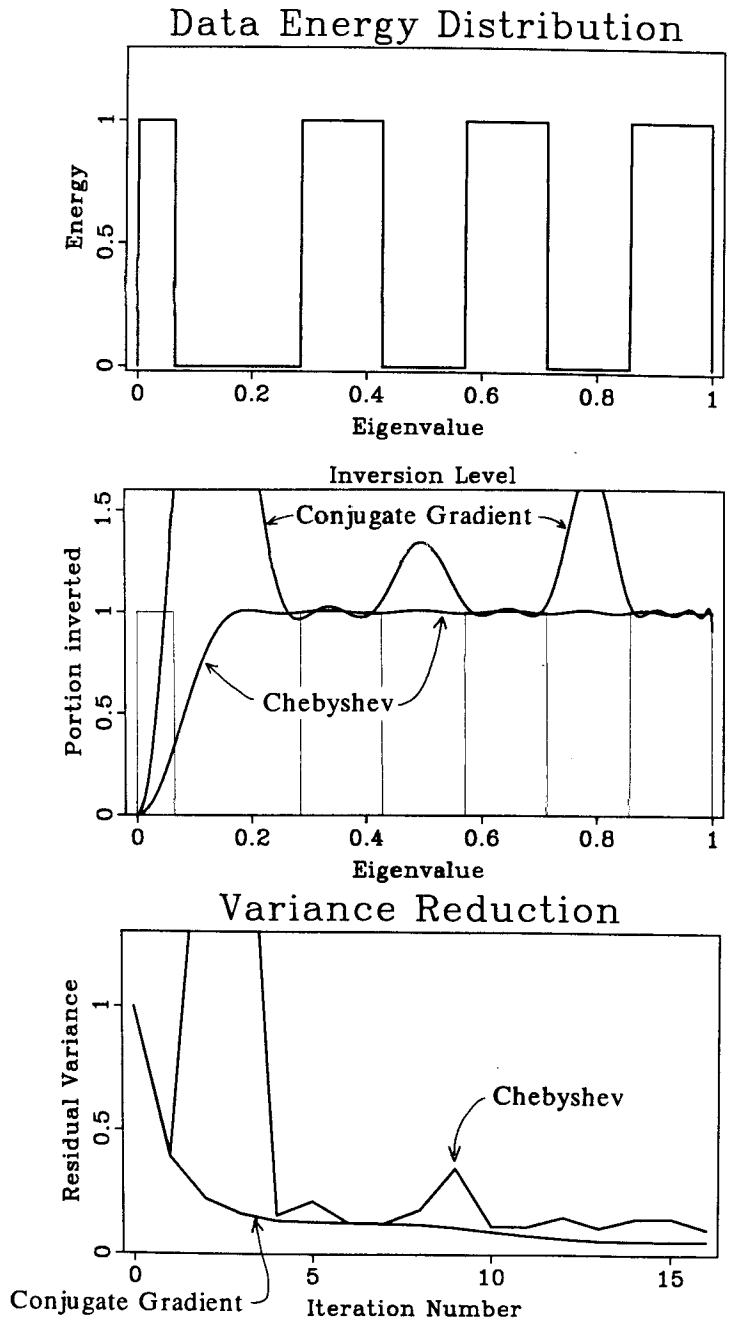
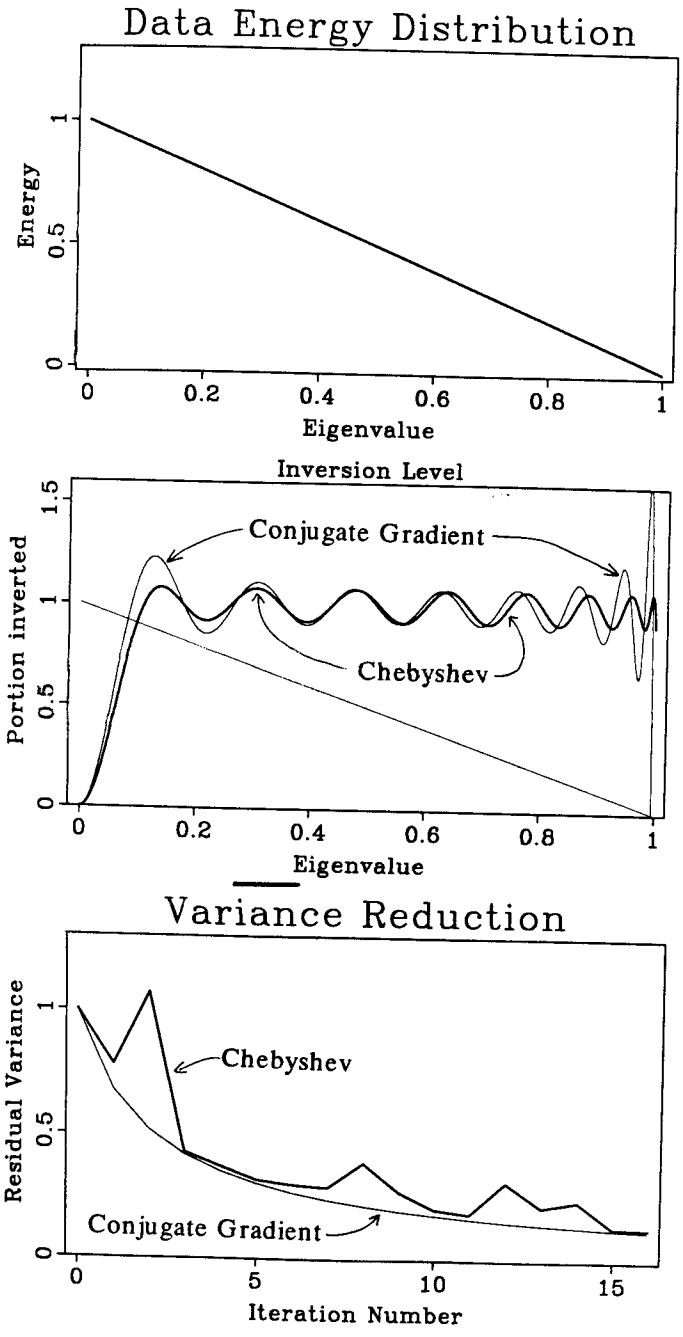


Figure 7: **a)** Data energy distribution weighted to the smaller eigenvalues **b)** Inversion level for conjugate gradient and Chebyshev iteration. Conjugate gradient has not inverted the very large eigenvalues very well because of their low energy levels. **c)** Variance reduction for both methods is similar, although conjugate gradient is still a little better.



An example of the maximum likelihood inversion is the stochastic inverse, which inverts the eigenvalue according to: $\lambda^G = \frac{\lambda}{\lambda^2 + \sigma}$. The smaller eigenvalues are less well inverted and their inversion level $(\lambda \cdot \lambda^G)$ is less than 1.0. The conjugate gradient inversion of Figure 4b has some of the inversion level of the smaller eigenvalues greater than 1.0. In particular, the peak at eigenvalue of 0.15 is quite large. This "overshoot" will introduce unwanted energy into the model.

A key drawback of conjugate gradient is that it is a "black box", one who's behavior is difficult to predict and cannot be controlled. One does not know whether an additional iteration will invert smaller eigenvalues or improve the accuracy of the inversion of the larger eigenvalue.

For instance, the inversion using banded data energy distribution in Figure 6b has inverted some of the eigenvalues near 0.05, which may not be warranted. One may instead want to more accurately invert the larger eigenvalues. Chebyshev iteration allows one to specify the eigenvalue range over which to invert and the accuracy to achieve over that range.

The control Chebyshev iteration gives a user over the inversion enables the him to use knowledge of the inverse problem to guide the inversion. He may have an objective that can be achieved with inversion to a specific eigenvalue. He would want an accurate inversion down to that eigenvalue, but not below.

Moreover, the availability of the control over the iterations may encourage the user to learn about the eigenvalue characteristics of his problem. The Chebyshev acceleration factors provide an excellent tool for the quantitative analysis of the problem through synthetics. Knowledge of the eigenvalue characteristics of the problem could then be directly used in the inversion. In addition, since Chebyshev iteration is not effected by the data energy distribution, a synthetic inversion can be exactly reproduced on the data.

However, the Chebyshev factors do not allow the luxury of starting the inversions having little knowledge of the problem's characteristics and being able to stop when convenient. When one doesn't know which range of eigenvalues to invert over, using Chebyshev scale factors can get cumbersome. After starting a series of iterations, one must continue until completion to analyze the results. If the results are unsatisfactory and another range of eigenvalues should be inverted, one must start over. One cannot backtrack or continue when using the Chebyshev factors, something possible with conjugate gradient.

In many applications such as the tomographic one, inversion should not be performed without familiarity of the characteristics of the intended application (Stork, 1988a). This familiarity is especially important for the proper interpretation of the

result. Several synthetics inversions should be performed as well as several data inversion with different parameters.

Without knowledge of the maximum eigenvalue, it is not possible to choose the Chebyshev scale factors. This requirement is not a problem for the Dines and Lytle (1979) back-projection formula since the data and model weights it imposes ensure that the maximum eigenvalue will be 1.0. In other situations where the maximum eigenvalue cannot be controlled through weighting, one can generally produce an accurate bounded estimate of the maximum eigenvalue by performing short test inversions of random noise. If the iterations diverge, the maximum eigenvalue is greater than estimated.

CONCLUSION

Although Richardson's iteration with Chebyshev scaling factors is slightly less efficient than conjugate gradient iteration, it produces a result closer to the maximum likelihood inversion, allows control over the inversion, and enables direct comparison of synthetics to data applications. However, determination of Chebyshev acceleration factors requires the maximum eigenvalue and the number of iterations be known before starting the iterations, which complicates the inversion.

ACKNOWLEDGEMENTS

This work is a direct continuation of thesis research performed under Rob Clayton at the Caltech Seismology Lab.

This work results from discussions with John Scales of Amoco. The author very much appreciates his feedback and encouragement.

I thank Joe Dellinger and Steve Cole and the rest of the present and past SEP members who have put together a very powerful and truly device independent graphics package.

REFERENCES

Comer, R.P. and Clayton, R.W., 1985, Reconstruction of mantle heterogeneity by iterative back-projection of travel times, 1: theory and reliability, preprint.

Dines, K.A. and Lytle, R.J., 1979, Computerized geophysical tomography: proceedings of the IEEE, **67**, 1065-1073.

Hestens, M. and Stiefel, E., 1952, Methods of conjugate gradients for solving linear systems, Nat. Bur. Standards J. Res., **49**, 409-436.

Ivansson, S., 1983. Remark on an earlier proposed iterative tomographic algorithm, Geophys. J. R. astr. Soc., **75**, 855-860.

Olson, A., 1986. A Chebyshev condition for accelerating convergence of iterative tomographic methods--solving large least squares problems, preprint.

Scales, J., 1987. Tomographic inversion via the conjugate gradient method, Geophysics, **52**, 179-185.

Stork, C., 1988a. Travel time tomographic velocity analysis of seismic surface reflection data, Ph.D. Thesis, Caltech

Stork, C., 1988b. Modification of conjugate gradient iteration to enable control of the eigenvalue range inverted, SEP-**57**.

Toldi, J., 1985. Velocity analysis without picking, Ph. D. Theses, Stanford University.

Appendix:

PART A: REPRESENTING THE VARIANCE IN TERMS OF EIGENVALUES

In the conventional form, variance is:

$$variance = \left| \left| \mathbf{b}^{(0)} - \mathbf{A}(\mathbf{A}^G \mathbf{b}^{(0)}) \right| \right|$$

This can be rewritten in terms of eigenvalues by using the Singular Value Decomposition representation for $\mathbf{A}$, $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$:

$$= \left| \left| \mathbf{b}^{(0)} - \mathbf{U}\Sigma\Sigma^G \mathbf{U}^T \mathbf{b}^{(0)} \right| \right|$$

Since the eigenvector matrix $\mathbf{U}$ is orthonormal ($\mathbf{U}\mathbf{U}^T = \mathbf{I}$), its introduction does not affect the variance:

$$= \left| \left| \mathbf{U}^T \left(\mathbf{b}^{(0)} - \mathbf{U}\Sigma\Sigma^G \mathbf{U}^T \mathbf{b}^{(0)} \right) \right| \right|$$

$$= \left| \left| \mathbf{U}^T \mathbf{b}^{(0)} - \Sigma\Sigma^G \mathbf{U}^T \mathbf{b}^{(0)} \right| \right|$$

The operation $\mathbf{U}^T \mathbf{b}^{(0)}$ separates the data into the components corresponding to each eigenvalue. The energy in the data corresponding to each eigenvalue is defined as:

$$E_i = \left((\mathbf{U}^T \mathbf{b}^{(0)})_i \right)^2$$

The data variance can be written as:

$$= \sum_i \mathbf{E}_i \cdot (1 - \lambda_i \cdot \lambda_i^G)^2$$

To treat the distribution of data energy as a continuum over the eigenvalue range, it is written as a function: $E(\lambda)$ where the distribution of eigenvalues is included in the function.

The variance can thus be written as:

$$variance = \int_{\lambda_{min}}^{\lambda_{max}} E(\lambda) \cdot (1.0 - \lambda \cdot \lambda^G)^2 \cdot d\lambda$$

where :
λ^G represents the generalized inversion of λ.
$E(\lambda)$ is the distribution of the data energy over the eigenvalue range.

In mathematical terms, $E(\lambda)$ is:

$$E(\lambda) \cdot d\lambda = \left(\lim_{d\lambda \to 0} \int_{\lambda}^{\lambda + d\lambda} \mathbf{b}^{(0)^T} \cdot \mathbf{u}(\lambda') \, d\lambda' \right)^2$$

where:
$\mathbf{u}(\lambda')$ = the data space eigenvector at eigenvalue λ' .

PART B: UNSUCCESSFUL DECOMPOSITION OF CONJUGATE GRADIENT ITERATION

An unsuccessful attempt is made to represent the result after n conjugate gradient iterations, $\mathbf{x}^{(n)}$, in terms of a polynomial in $\mathbf{A}^T\mathbf{A}$, such that:

$$\mathbf{x}^{(n)} = \left(\sum_{i=0}^{n} \beta_i \cdot (\mathbf{A}^T\mathbf{A})^i \right) \cdot \mathbf{A}^T\mathbf{b}^{(0)}$$

This result would enable the determination of:

$$\lambda^G = \sum_{i=0}^{n} \beta_i' \cdot (\lambda)^i$$

as was done for Richardson's iterations.

The attempt is made by substituting in recursively for $\mathbf{x}^{(i)}$ until the function for $\mathbf{x}^{(n)}$ is only in terms of $\mathbf{b}^{(0)}$ and $\mathbf{x}^{(0)}$. $\mathbf{x}^{(0)}$ is defined to be equal to zero, giving us the function in terms of $\mathbf{b}^{(0)}$.

After two substitutions, no recursion pattern could be identified as was for Richardson's iteration and the result was too complicated to continue. However, it is clear that $\mathbf{x}_{(n)}$ can be represented by a polynomial in $\mathbf{A}^T\mathbf{A}$.

$$\mathbf{x}^{(n)} = \mathbf{x}^{(n-1)} + \sigma_n \cdot \mathbf{p}_n$$

$$\mathbf{p}_n = \mathbf{A}^T\mathbf{b}^{(n)} + \sigma_n \cdot \mathbf{p}_{n-1}, \quad \mathbf{p}_0 = 0$$

$$\mathbf{p}_n = \sum_{i=0}^{n} \left[\prod_{j=i+1}^{n} \mu_j \right] \cdot \mathbf{A}^T\mathbf{b}^{(i)}$$

$$\mathbf{b}^{(i)} = \mathbf{b}^{(0)} - \mathbf{A}\mathbf{x}^{i-1}$$

$$\mathbf{x}^{(n)} = \mathbf{x}^{(n-1)} + \sigma_n \cdot \left(\sum_{i=0}^{n} \left[\prod_{j=i+1}^{n} \mu_j \right] \mathbf{A}^T \cdot \mathbf{b}^{(i)} \right)$$

$$\mathbf{x}^{(n)} = \mathbf{x}^{(n-2)} + \sigma_{n-1} \cdot \left(\sum_{i=0}^{n-1} \left[\prod_{j=i+1}^{n-1} \mu_j \right] \mathbf{A}^T \cdot \mathbf{b}^{(i)} \right) +$$

$$\sigma_n \cdot \left(\sum_{i=0}^{n-1} \left[\prod_{j=i+1}^{n} \mu_j \right] \mathbf{A}^T \cdot \mathbf{b}^{(i)} \right) +$$

$$\sigma_n \cdot \left(\mathbf{A}^T \cdot (\mathbf{b}^{(0)} - \mathbf{A}\mathbf{x}^{(n-1)}) \right)$$

$$\mathbf{x}^{(n)} = \mathbf{x}^{(n-2)} + \sigma_{n-1} \cdot \left(\sum_{i=0}^{n-1} \left[\prod_{j=i+1}^{n-1} \mu_j \right] \mathbf{A}^T \cdot \mathbf{b}^{(i)} \right) +$$

$$\sigma_n \cdot \left(\sum_{i=0}^{n-1} \left[\prod_{j=i+1}^{n} \mu_j \right] \mathbf{A}^T \cdot \mathbf{b}^{(i)} \right) +$$

$$\sigma_n \cdot \left(\mathbf{A}^T \cdot \left(\mathbf{b}^{(0)} - \mathbf{A}\mathbf{x}^{n-2} \right) \right) +$$

$$- \sigma_n \cdot \mathbf{A}^T \mathbf{A} \cdot \sigma_{n-1} \left(\sum_{i=0}^{n-1} \left[\prod_{j=i+1}^{n-1} \mu_j \right] \mathbf{A}^T \cdot \mathbf{b}^{(i)} \right)$$

$$\mathbf{x}^{(n)} = \mathbf{x}^{(n-2)} + \sigma_{n-1} \cdot \left(\sum_{i=0}^{n-1} \left[\prod_{j=i+1}^{n-1} \mu_j \right] \mathbf{A}^T \cdot \mathbf{b}^{(i)} \right) +$$

$$\sigma_n \cdot \left(\sum_{i=0}^{n-1} \left[\prod_{j=i+1}^{n} \mu_j \right] \mathbf{A}^T \cdot \mathbf{b}^{(i)} \right) +$$

$$\sigma_n \mathbf{A}^T \mathbf{b}^{(0)} - \sigma_n \mathbf{A}^T \mathbf{A}\mathbf{x}^{(n-2)} -$$

$$\sigma_n \mathbf{A}^T \mathbf{A} \sigma_{n-1} \left(\sum_{i=0}^{n-1} \left[\prod_{j=i+1}^{n-1} \mu_j \right] \mathbf{A}^T \cdot \mathbf{b}^{(i)} \right)$$

$$\mathbf{x}^{(n)} = \left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A} \right) \mathbf{x}^{(n-2)} +$$

$$\left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A} \right) \cdot \sigma_{n-1} \cdot \left(\sum_{i=0}^{n-1} \left[\prod_{j=i+1}^{n-1} \mu_j \right] \mathbf{A}^T \cdot \mathbf{b}^{(i)} \right) +$$

$$\sigma_n \cdot \left(\sum_{i=0}^{n-1} \left[\prod_{j=i+1}^{n} \mu_j \right] \mathbf{A}^T \mathbf{b}^{(i)} \right) +$$

$$\sigma_n \mathbf{A}^T \mathbf{b}^{(0)}$$

$$\mathbf{x}^{(n)} = \left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A} \right) \mathbf{x}^{(n-2)} +$$

$$\left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A} \right) \cdot \sigma_{n-1} \cdot \left(\sum_{i=0}^{n-1} \left[\prod_{j=i+1}^{n-1} \mu_j \right] \mathbf{A}^T \cdot \mathbf{b}^{(i)} \right) +$$

$$\sigma_n \cdot \mu_n \cdot \left(\sum_{i=0}^{n-1} \left[\prod_{j=i+1}^{n-1} \mu_j \right] \mathbf{A}^T \mathbf{b}^{(i)} \right) +$$

$$
\begin{aligned}
&\sigma_n \mathbf{A}^T \mathbf{b}^{(0)} \\
\mathbf{x}^{(n)} = &\left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A}\right)\left(\mathbf{I} - \sigma_{n-1}\mathbf{A}^T \mathbf{A}\right)\mathbf{x}^{(n-3)} + \\
&\left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A}\right)\left(\mathbf{I} - \sigma_{n-1}\mathbf{A}^T \mathbf{A}\right)\cdot\sigma_{n-2}\cdot\left(\sum_{i=0}^{n-2}\left[\prod_{j=i+1}^{n-2}\mu_j\right]\mathbf{A}^T\cdot\mathbf{b}^{(i)}\right) + \\
&\left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A}\right)\cdot\sigma_{n-1}\cdot\mu_{n-1}\cdot\left(\sum_{i=0}^{n-2}\left[\prod_{j=i+1}^{n-2}\mu_j\right]\mathbf{A}^T \mathbf{b}^{(i)}\right) + \\
&\left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A}\right)\sigma_{n-1}\mathbf{A}^T \mathbf{b}^{(0)} \\
&\sigma_n \mathbf{A}^T \mathbf{b}^{(0)} \\
&\sigma_n \cdot\mu_n \cdot\left(\sum_{i=0}^{n-2}\left[\prod_{j=i+1}^{n-2}\mu_j\right]\mathbf{A}^T \mathbf{b}^{(i)}\right) + \\
&\sigma_n \cdot\mu_n \cdot\left(\mathbf{A}^T\left(\mathbf{b}^{(0)} - \mathbf{A}\mathbf{x}^{n-2}\right)\right) + \\
\mathbf{x}^{(n)} = &\left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A}\right)\left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A}\right)\mathbf{x}^{(n-3)} + \\
&\left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A}\right)\left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A}\right)\cdot\sigma_{n-2}\cdot\left(\sum_{i=0}^{n-2}\left[\prod_{j=i+1}^{n-2}\mu_j\right]\mathbf{A}^T\cdot\mathbf{b}^{(i)}\right) + \\
&\left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A}\right)\cdot\sigma_{n-1}\cdot\mu_{n-1}\cdot\left(\sum_{i=0}^{n-2}\left[\prod_{j=i+1}^{n-2}\mu_j\right]\mathbf{A}^T \mathbf{b}^{(i)}\right) + \\
&\left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A}\right)\sigma_{n-1}\mathbf{A}^T \mathbf{b}^{(0)} \\
&\sigma_n \mathbf{A}^T \mathbf{b}^{(0)} \\
&\sigma_n \cdot\mu_n\, \mu_{n-1}\cdot\left(\sum_{i=0}^{n-2}\left[\prod_{j=i+1}^{n-2}\mu_j\right]\mathbf{A}^T \mathbf{b}^{(i)}\right) + \\
&\sigma_n \cdot\mu_n \cdot\left(\mathbf{A}^T\left(\mathbf{b}^{(0)} - \mathbf{A}\mathbf{x}^{n-3}\right)\right) + \\
&-\sigma_n \cdot\mu_n \cdot\left(\mathbf{A}^T \mathbf{A}\sigma_{n-2}\cdot\left(\sum_{i=0}^{n-2}\left[\prod_{j=i+1}^{n-2}\mu_j\right]\mathbf{A}^T\cdot\mathbf{b}^{(i)}\right)\right) + \\
\mathbf{x}^{(n)} = &\left(\mathbf{I} - \varrho_n \mathbf{A}^T \mathbf{A}\right)\left(\mathbf{I} - \sigma_{n-1}\mathbf{A}^T \mathbf{A}\right)\mathbf{x}^{(n-3)} + \\
&-\sigma_n \cdot\mu_n \cdot\mathbf{A}^T \mathbf{A}\mathbf{x}^{n-3} + \\
&\left(\mathbf{I} - \sigma_n \mathbf{A}^T \mathbf{A}\right)\left(\mathbf{I} - \sigma_{n-1}\mathbf{A}^T \mathbf{A}\right)\cdot\sigma_{n-2}\cdot\left(\sum_{i=0}^{n-2}\left[\prod_{j=i+1}^{n-2}\mu_j\right]\mathbf{A}^T\cdot\mathbf{b}^{(i)}\right) +
\end{aligned}
$$

$$\left(\mathbf{I}-\sigma_n\mathbf{A}^T\mathbf{A}\right)\cdot\sigma_{n-1}\cdot\mu_{n-1}\cdot\left(\sum_{i=0}^{n-2}\left[\prod_{j=i+1}^{n-2}\mu_j\right]\mathbf{A}^T\mathbf{b}^{(i)}\right)+$$

$$\left(\mathbf{I}-\sigma_n\mathbf{A}^T\mathbf{A}\right)\sigma_{n-1}\mathbf{A}^T\mathbf{b}^{(0)}+\sigma_n\mathbf{A}^T\mathbf{b}^{(0)}+\sigma_n\cdot\mu_n\cdot\mathbf{A}^T\mathbf{b}^{(0)}$$

$$\left(\mu_{n-1}\mathbf{I}-\sigma_{n-2}\mathbf{A}^T\mathbf{A}\right)\sigma_n\cdot\mu_n\cdot\left(\sum_{i=0}^{n-2}\left[\prod_{j=i+1}^{n-2}\mu_j\right]\mathbf{A}^T\mathbf{b}^{(i)}\right)+$$

PART C: DETERMINATION OF RICHARSON'S SCALING FACTORS USING CHEBYSHEV POLYNOMIALS

This development shows how the Chebyshev polynomials can be used to find the scaling factors for Richardson's iteration such that $\lambda^G \approx \frac{1}{\lambda}$ over a predefined eigenvalue range.

Richarson's iteration with scaling factors of σ is written as:

$$\mathbf{x}^{(n+1)}=\mathbf{x}^{(n)}+\sigma_n\cdot\mathbf{A}^T\mathbf{b}^{(n)};$$

where:
$\mathbf{b}^{(n)}=\mathbf{b}^{(0)}-\mathbf{A}\mathbf{x}^{(n-1)}$
decreasing the superscripts by one and substituting for $\mathbf{b}^{(n)}$:

$$\mathbf{x}^{(n)}=\mathbf{x}^{(n-1)}+\sigma_n\mathbf{A}^T(\mathbf{b}^{(0)}-\mathbf{A}\mathbf{x}^{(n-1)});$$

rearranging:

$$\mathbf{x}^{(n)}=\sigma_n\mathbf{A}^T\mathbf{b}^{(0)}+(\mathbf{I}-\sigma_n\mathbf{A}^T\mathbf{A})\mathbf{x}^{(n-1)};$$

Substituting in recursively:

$$\mathbf{x}^{(n)}=\sigma_n\mathbf{A}^T\mathbf{b}^{(0)}+(\mathbf{I}-\sigma_n\mathbf{A}^T\mathbf{A})\sigma_{(n-1)}\mathbf{A}^T\mathbf{b}^{(0)}$$

$$+(\mathbf{I}-\sigma_n\mathbf{A}^T\mathbf{A})(\mathbf{I}-\sigma_{n-1}\mathbf{A}^T\mathbf{A})\mathbf{x}^{(n-2)}$$

$$=\sum_{l=0}^{n}\prod_{j=l+1}^{n}(\mathbf{I}-\sigma_j\mathbf{A}^T\mathbf{A})\ \sigma_l\mathbf{A}^T\mathbf{b}^{(0)}$$

$$+ \prod_{l=0}^{n} (\mathbf{I} - \sigma_l \mathbf{A}^T \mathbf{A}) \ \mathbf{x}^{(0)} .$$

$\mathbf{x}^{(0)}$, the starting point, is defined to be 0. Thus,

$$\mathbf{x}^{(n)} = \sum_{m=0}^{n-1} \prod_{j=0}^{m-1} \left(\mathbf{I} - \sigma_j \mathbf{A}^T \mathbf{A} \right) \sigma_m \mathbf{A}^T \mathbf{b}^{(0)}$$

which can be converted into independent linear equations for each eigenvalue by substituting in the singular decomposition for $\mathbf{A}$, $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$:

$$\mathbf{x}^{(n)} = \mathbf{V}\Sigma^G \mathbf{U}^T \mathbf{b}$$

where,

$$\Sigma^G = \sum_{m=0}^{n-1} \prod_{j=0}^{m-1} \left(\mathbf{I} - \sigma_j \Sigma^2 \right) \sigma_m \Sigma$$

Since Σ is a diagonal matrix of the eigenvalues, the independent equations are:

$$\lambda_i{}^G = \sum_{m=0}^{n-1} \sigma_m \prod_{j=0}^{m-1} (1 - \sigma_j \lambda_i^2) \ \lambda_i .$$

The equation can be rearranged by adding (1 – 1):

$$\lambda_i{}^G = \sum_{m=0}^{n-1} \lambda_i^{-2} \left(1 - (1 - \sigma_m \lambda_i^2) \right) \prod_{j=0}^{m-1} (1 - \sigma_j \lambda_i^2) \lambda_i$$

$$= \lambda_i^{-2} \sum_{m=0}^{n-1} \left(\prod_{j=0}^{m-1} (1 - \sigma_j \lambda_i^2) - \prod_{j=0}^{m} (1 - \sigma_j \lambda_i^2) \right) \lambda_i .$$

Most terms cancel each other out, leaving only:

$$\lambda_i{}^G = \frac{1 - \prod_{j=0}^{n-1}(1 - \sigma_j \lambda_i^2)}{\lambda_i}.$$

The objective is to have $\lambda_i{}^G \approx \frac{1}{\lambda_i}$ over the desired eigenvalue range, $\lambda_{\min} < \lambda_i < \lambda_{\max}$. This is best achieved by minimizing:

$$\prod_{j=0}^{n-1}(1 - \sigma_j \lambda_i^2) \qquad \text{for: } \lambda_{\min} < \lambda_i < \lambda_{\max}.$$

By relating this equation to the Chebyshev polynomials, Olson (1987), determines the optimal scale factors, σ_j so that the maximum value of the above polynomial is roughly even over a specified eigenvalue range. They are:

$$\sigma_n = \frac{2}{\cos\left(\frac{(2n+1)\pi}{2 \cdot N}\right) \cdot (\lambda_{\max}^2 - \lambda_{\min}^2) + (\lambda_{\max}^2 + \lambda_{\min}^2)}$$