RECEIVED BY OSTI
FEB 13 1990

TITLE Acceleration of Convergence

AUTHOR(S): Lawrence Auer

SUBMITTED TO: NATO Advanced Workshop, Stellar Atmospheres: Beyond
Classical Models, September 3-7, 1990, Trieste, Italy

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

By acceptance of this article, the publisher recognizes that the U S Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U S Government purposes.

The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U S Department of Energy

MASTER
Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

Acceleration of Convergence

Lawrence Auer

EES-5

Los Alamos National Laboratory

ABSTRACT. Acceleration of the convergence of approximate operator iteration schemes is discussed. Algorithms based on both residual minimization and use of conjugate vector spaces are presented. It is shown that both give dramatic improvement, at very low computational cost, in the iterative solution of radiative transfer problems in the presence of scattering.

I. INTRODUCTION

It is a fundamental (and perverse) law of nature that it is easy to find simple but slowly converging iterative methods to solve systems of equations. With a lot of work and luck it is often possible to find clever methods for efficiently solving specific problems. In this paper, however, we will take the simpler approach, and introduce a class of general methods that can be used to speed up the convergence of any linearly convergent method. Acceleration can improve convergence so dramatically that even very simple iteration schemes become acceptably efficient. The methods that will be introduced are robust, generally applicable, and need only a sequence of estimates of the solution, the convergence of which is what one is trying to accelerate.

II. Approximate Operator Iteration

In many cases it is impractical to solve directly the set of linear equations, $\mathbf{Ax} = b$, either because of the size of matrix $\mathbf{A}$ or because its actual value is not efficiently calculable. It is, nevertheless, often still efficient to compute $\mathbf{Ax}$, which makes iterative solution effective.

Short characteristic formal solution of the transfer equation is an important example, Kunasz and Auer (1988). Even in the three dimensional case the transfer equation may be rapidly solved for known source function. The time for evaluation is proportional to the number of points. If we try to solve the scattering problem directly, the computing time scales as N^3 , while iterative solution scales only as $m_{iter} N^3$. As long as we can keep the number of iterations, m_{iter} , reasonable, the relative advantage of iterative solution is dramatic.

Iterative methods are created by replacing the true matrix $\mathbf{A}$ by an approximation, $\mathbf{A} = \mathbf{A}_* + \Delta\mathbf{A}$, where $\mathbf{A}_*$ is chosen such that solution the equations, $\mathbf{A}_*\mathbf{x} = b$, is feasible. If $\mathbf{x}_i$ is the current estimate of the solution, the next estimate may be written

$$\begin{aligned}\vec{x}_{i+1} &= \vec{x}_i + \Delta\vec{x} \quad \text{and} \\ \mathbf{A}_*\Delta\vec{x} &= \vec{b} - \mathbf{A}\vec{x}_i + O(\Delta\mathbf{A}\Delta\vec{x})\end{aligned}$$

One iteration, thus, consists of

$$\begin{aligned}\Delta \vec{x} &= \mathbf{A}_*^{-1}(\vec{b} - \mathbf{A}\vec{x}) \\ \vec{x} &\leftarrow \vec{x} + \Delta \vec{x}\end{aligned}\quad (1)$$

The use of an approximate operator has functionally replaced the original system by one with the same solution,

$$\mathbf{A}_*^{-1} \mathbf{A} \vec{x} = \mathbf{A}_*^{-1} \vec{b}, \quad (2)$$

which may be written

$$\mathbf{A}' \vec{x} = \vec{b}'$$

which is why $\mathbf{A}_*$ is often referred to a preconditioner. Note, the residual of the preconditioned equations is just the change in $\vec{x}$ which gives the next estimate, i.e., $\Delta \vec{x}$.

Written in terms of $\vec{x}$ rather than $\Delta \vec{x}$, the iterative cycle is

$$\vec{x}_{i+1} = (\mathbf{I} - \mathbf{A}_*^{-1} \mathbf{A}) \vec{x}_i + \mathbf{A}_*^{-1} \vec{b} \quad (3)$$

so one wants

$$\mathbf{A}_*^{-1} \mathbf{A} \approx \mathbf{I}$$

in order to minimize the need for iteration. It is easy to show how Eq. (3) converges. If $\vec{x}$ is the converged solution, each estimate may be written

$$\vec{x}_i = \vec{x} + \vec{e}_i,$$

where $\vec{e}_i$ is the error of the i -th iterate. Expanding the error in terms of the eigenvectors of the iteration matrix, we obtain

$$\vec{e}_n = \mathbf{F} \vec{e}_{n-1} = \sum_j \lambda_j^n a_j \vec{v}_j, \quad (4)$$

which means that asymptotically each iteration decreases the error by a factor of λ_{max} . Any iterative method which uses only information from the previous iteration cannot converge faster than linearly. If the maximum eigenvalue is only slightly less than unity, convergence will be painfully slow and, thus, one has a need for acceleration.

III. THE CLASSIC EXAMPLE: LAMBDA ITERATION

In radiative transfer the most frequently occurring example of poor convergence is an attempt to use the Lambda operator and the definition of the source function to iterate for $S(\tau)$. For example, consider a complete redistribution problem,

$$S = (1 - \epsilon) \Lambda(S) + \epsilon B$$

in a semi-infinite atmosphere. Convergence is controlled by the largest eigenvalue of $I - (1 - \epsilon)\Lambda$. Using complete redistribution and 7 points per decade to resolve the outer section of a semi-infinite atmosphere, the largest eigenvalue is 0.9999318, which yields the results shown here in Fig 1. (A parenthetic remark about using ΔS to judge convergence is appropriate: **BE CAREFUL!** Just because the iterative scheme gives smaller and smaller changes which eventually appear to be “negligible”, one should not assume that one has converged. This is exactly the behavior shown in Fig 1 and is due entirely to reduction of the eigenvectors with small eigenvalues. For the Lambda iteration a reasonable stopping criterion is that the maximum $\Delta S/S \ll \epsilon$, as $1 - \epsilon$ is a good upper bound to the convergence rate.)

Figure 1. Twenty Lambda iterations in semi-infinite atmosphere, complete redistribution over a Doppler profile, $\epsilon = 10^{-3}$. Converged solution marked by*.

Figure 2. Twenty iterations using Scharmer's multi-point approximate operator.

The most desirable solution to the “Lambda Iteration Problem” is to find a good approximate operator. Scharmer (1981) has proposed an excellent one based on the Eddington-Babier relation. Its convergence is shown in Fig 2. Although this approximate operator has very good convergence, there are problems. It is not trivial to generate and it does not generalize to multi-dimensional problems. To avoid these difficulties we are forced to sacrifice the speed of convergence and use a diagonal approximate operator. The inverse of such an operator is just the scalar inverse of the elements. Olsen, Auer and Buchler (1986), hereafter OAB, demonstrated the utility of the diagonal the Lambda matrix as the approximate operator. The convergence for this choice is shown in Fig 3. While it is significantly better than straight Lambda iteration, it is still relatively slow. If we accelerate it using third order conjugate vector acceleration, for example,

the convergence is more than adequate, as is shown in Fig 4. Essentially the same rate of convergence is obtained for multi-dimensional cases, which is of extreme importance given the ease of construction of the OAB one point operator.

Figure 3. Convergence with a simple one-point operator, the diagonal of the Λ -matrix.

Figure 4. Acceleration of the convergence of the OAB operator using three conjugate vectors (ORTH3).

IV. RESIDUAL MINIMIZATION

The two classes of acceleration techniques that we will describe are 1) residual minimization and 2) minimization with respect to a set of conjugate vectors. The basis of both techniques is the observation that the residual contains the error. That is, the residual is not orthogonal to the error, and the error can be approximated by a linear combination of those residuals.

The first approach is based on the idea of milking what you have. Using approximate operate iteration we generate the iterates $x_i, x_{i+1}, \dots, x_{i+m}$, with "residuals" $r_i, r_{i+1}, \dots, r_{i+m}$. The converged solution corresponds to $r=0$. We accordingly we seek a linear combination of the estimates which reduces the residual. If that combination is written

$$\vec{x} = \left(1 - \sum_{j=0}^{m-1} a_j\right) \vec{x}_{i+m} + \sum_{j=0}^{m-1} a_j \vec{x}_{i+j}, \quad \text{the residual is } \vec{r} = \left(1 - \sum_{j=0}^{m-1} a_j\right) \vec{r}_{i+m} + \sum_{j=0}^{m-1} a_j \vec{r}_{i+j},$$

which is minimized by solving the weighted least squares problem for $r \approx 0$. The normal equations for a_i are $Ca = b$, where

$$c_{jk} = \sum \bar{w}(\bar{r}_{i+j} - \bar{r}_{i+m})(\bar{r}_{i+k} - \bar{r}_{i+m}) \quad \text{and} \\ b_j = \sum \bar{w}\bar{r}_{i+m}(\bar{r}_{i+j} - \bar{r}_{i+m}) \quad . \quad (5)$$

Summation is over the length of the vectors and w is the weights. The “residuals” used in these equations are actually the changes in the successive iterates, i.e., $r_k = \Delta x_k = x_{k+1} - x_k$, which are the residuals of the *preconditioned* equations, Eq. (2). The convergence has been found to be improved by using weights which are inversely proportional to size of the element. This corresponds to minimizing the relative error.

The number of coefficients we use in the acceleration, n , affects the rate of convergence, as shown in Figs 5a,b. Although these figures seem to imply a clear superiority for larger n , in fact, the asymptotic convergence is not actually dramatically improved by going to larger n , as seen in Fig. 7. Further, before one can accelerate again one needs to make at least $n + 1$ iterations. This means that high order acceleration leads to a “crawl-jump” convergence which can require extra iterations just to know you are actually done. The utility of low order extrapolation is particularly significant because it is necessary to provide temporary storage for $n + 2$ arrays the size of the system. The availability of storage may be a critical factor when dealing with the very large systems required when dealing with multi-dimensional problems.

Figure 5. Acceleration using residual minimization a) $n=2$, and b) $n=4$. Both are fully converged in less than twenty iterations.

The first use of residual minimization to accelerate the convergence of an approximate operator approach to radiative transfer was made by Olsen, et al. (1986). It is described in greater detail in Auer (1987) who gives pseudocode for the algorithm. The basic ideas apparently were developed by a number of independent workers, Ng (1974)

(1974) and Messina (1977). Reviews and comparison of related residual minimization techniques are given in the articles by Side et al. (1986) and Smith et al. (1987).

V. MINIMIZATION WITH RESPECT TO A SET OF CONJUGATE VECTORS

A more mathematically sophisticated approach is based on residual minimization with respect to a set of *conjugate* vectors rather than just with respect to the residuals themselves. The mathematical nature of the vectors is seen from the following considerations. Say the residual of our preconditioned system is $\vec{r}_i = b - \mathbf{A}\vec{x}_i$. We can add a vector, $\vec{u}$, to reduce this residual in the least squares sense

$$\vec{x}_{i+1} = \vec{x}_i + a\vec{u}, \quad \text{where} \quad a = \frac{\vec{r}_i \cdot \mathbf{A}\vec{u}}{\mathbf{A}\vec{u} \cdot \mathbf{A}\vec{u}} \equiv \frac{(\vec{r}_i, \mathbf{A}\vec{u})}{(\mathbf{A}\vec{u}, \mathbf{A}\vec{u})}.$$

The new residual is

$$\vec{r}_{i+1} = \vec{r}_i - a\mathbf{A}\vec{u}.$$

If we correct again

$$\vec{x}_{i+2} = \vec{x}_i + a\vec{u} + b\vec{w}, \quad \text{with} \quad b = \frac{(\vec{r}_{i+1}, \mathbf{A}\vec{w})}{(\mathbf{A}\vec{w}, \mathbf{A}\vec{w})}.$$

But

$$(\vec{r}_{i+1}, \mathbf{A}\vec{w}) = (\vec{r}_i, \mathbf{A}\vec{w}) - a(\mathbf{A}\vec{u}, \mathbf{A}\vec{w}),$$

so unless $(\mathbf{A}\vec{u}, \mathbf{A}\vec{w}) = 0$, we are UNDOING part of what we just did! The set of conjugate vectors, which when multiplied by $\mathbf{A}$ are orthogonal, spans the space most efficiently.

How does one create the set of conjugate vectors? As observed above the residuals of each iteration are "rich" in what we want to remove, the error; therefore, we can use the residual, $\vec{r}_i$, as the basis of the next conjugate vector, $\vec{p}_i$. All we need to do is to make the residual orthogonal to the previously found $\vec{p}_k$. This is done by

$$\vec{p}_i = \vec{r}_i - \sum_k b_k \vec{p}_k, \quad (6a)$$

and

$$\mathbf{A}\vec{p}_i = \mathbf{A}\vec{r}_i - \sum_k b_k \mathbf{A}\vec{p}_k, \quad \text{with} \quad b_k = \frac{(\vec{r}_i, \mathbf{A}\vec{p}_k)}{(\mathbf{A}\vec{p}_k, \mathbf{A}\vec{p}_k)} \quad (6b)$$

If we limit the size of the conjugate vector basis set to be n vectors, we need $2n+2$ scratch storage vector to hold the $\vec{p}_k$, $\mathbf{A}\vec{p}_k$, and the current residual and extrapolant.

In presenting the concepts of conjugate vectors we have assumed that $\mathbf{A}\vec{r}$ as well as $\vec{r}$ was available, although in the introductory remarks we stated that we should not need to know the matrix explicitly. Fortunately, both of these vectors may be constructed from three successive iterates,

$$\begin{aligned}\vec{r} &= \vec{x}_i - \vec{x}_{i-1} \\ \mathbf{A}' \vec{r} &= r - (\vec{x}_{i+1} - \vec{x}_i) \end{aligned} \quad (7)$$

The first relation follows from the fact that the "residual" of the preconditioned system is just the Δx predicted. The second relation can be shown by observing

$$\begin{aligned}\vec{x}_{i+1} &= \vec{x}_i + \vec{b}' - \mathbf{A}' \vec{x}_i \\ &= \vec{x}_i + \vec{b}' - \mathbf{A}' (\vec{x}_{i-1} + \vec{r}) \\ &= \vec{x}_i + \vec{r} - \mathbf{A}' \vec{r} \end{aligned}$$

We should note that because the x_j in the above equations are produced by the approximate operator iteration, the $\mathbf{A}'$ and $\vec{b}'$ appearing in the above equations are the preconditioned terms from Eq. (2). Further, Eq. (7) applies even if each iterate is accelerated, as long as we use $x_i - x_{i-1}^{accel}$ and $x_{i+1} - x_i^{accel}$, where x is the ordinary next iterate predicated using x^{accel} .

We simply remove the new p and $\mathbf{A}' p$ from r in order to accelerate the convergence. The improved estimate of solution is

$$\begin{aligned}\vec{x}_{i+1}^{accel} &= \vec{x}_i + a(\vec{p} - \mathbf{A}' \vec{p}) \\ a &= \frac{(\vec{r}, \mathbf{A}' \vec{p})}{(\mathbf{A}' \vec{p}, \mathbf{A}' \vec{p})} \end{aligned} \quad (8)$$

Because of orthogonality we do not have to reconsider other p_k . Eq. (8) implicitly reduces r by adding p to x_{i-1} which in turn affects the next estimate. Its derivation proceeds:

$$\begin{aligned}\vec{x}' &= x_{i-1} + a\vec{p} \\ \vec{r}' &= \vec{r} - a\mathbf{A}' \vec{p} \\ \vec{x}^{accel} &= \vec{x}' + \vec{r}' \end{aligned}$$

Practically, one has to limit the number of p_k and $\mathbf{A}' p_k$ that are retained. If the number of iterations exceeds the storage available, then the oldest vectors are overwritten. To keep information on n back p_k , and $\mathbf{A}' p_k$ plus the x and r from the last iteration requires space for $2n+2$ scratch arrays. The value $n = 1$ should never be used, and larger values uniformly improve the speed of convergence. Fortunately, for the radiative transfer equation $n=2$ is entirely adequate. This may not be true for less inherently symmetric systems.

The algorithm proceeds as follows initialization, saving x_2 and r_1 :

On every subsequent iteration

If at storage limit, discard oldest p_k, Ap_k

Eq 7, to compute r and Ar

Eq 6, to compute newest p, Ap

Eq 8, to compute accelerated x , which is returned

Store p, Ap, r , and x internally

Regardless of the number of back vectors being saved, acceleration is performed every time a new p is found, that is every iteration after the second one. An example of the convergence achieved using $n=2$ and $n=4$ on our standard problem is shown in Fig 6a,b. The most striking difference between this and the results of residual minimization is the relative smoothness of the accelerated convergence.

Figure 6. Acceleration using residual minimization with respect to a set of conjugate vectors, a) $n=2$, and b) $n=4$.

The basic conjugate vector acceleration technique was introduced by Vinsome (1976) in his implementation of the ORTHOMIN routine. Its first application to the acceleration of radiative transfer calculations was Klein, et al. (1989). There are actually a number of ways in which the concept may be implemented. A good introduction to them is Eisenstadt et al. (1983) and Saad and Schultz (1985, 1986)

VI. CONCLUSIONS

Including either residual minimization (RM) or conjugate vector acceleration (ORTH) in a running iterative code is very simple. Both only require the current estimate as input, and replace it with the accelerated value as output. The basic iteration loop is simply

Figure 7. Comparison of the convergence of the maximum relative error for various acceleration methods and orders. The residual minimization (RM) and conjugate vector (ORTH) have been used to accelerate the convergence of the OAB operator.

```

loop
  Approximate Operator => next Estimate
  Acceleration(Estimate) => better Estimate
until Converged

```

The internal operation of the acceleration routines is totally transparent to the calling routine. Inclusion requires only allocation of scratch storage and a single call. Fortran implementations of both algorithms are available on request from the author.

The comparative behavior of the methods is shown in Fig. 7 for the $\epsilon = 10^{-3}$ problem. The results seen here are typical. Straight Lambda iteration yields the nearly horizontal, slowly (!!!) converging, line. Use of the diagonal of the Lambda as an approximate operator improves convergence, and is shown in the unaccelerated line. Remarkably for both ORTH and RM the low order method, $n=2$, yields effective acceleration. For residual minimization the RM2 case is definitely preferable to $n=4$. While the latter may occasionally yield better estimates, its convergence is so jumpy that one cannot take advantage of this in general. For conjugate vector acceleration, on the other hand, higher order gives significant improvement. The convergence rates shown by ORTH2 and

RM2 are reasonably similar, with ORTH2 slightly faster at the cost of storage for two more scratch arrays. Both are reasonable choices in the absence of other information. If memory is not a consideration, ORTH may be preferable because of the smoothness of convergence and the effectiveness of higher order acceleration. The optimal choice of order and method has to be made on the basis of direct testing, as these depend on the details of the eigenspectrum of the iteration scheme being used.

ACKNOWLEDGMENT

This work has been supported by Los Alamos National Laboratory which is operated by the University of California for the U. S. Department of Energy.

REFERENCES

- Auer, L.H. (1987) "Acceleration of Convergence", in *Numerical Radiative Transfer*, ed W. Kalkofen, Cambridge University Press, Cambridge.
- Eisenstat, S.C., Eleman, H.C., and Schultz, M.R. (1983) "Variational Iterative Methods for Nonsymmetric Systems of Linear Equations", *SIAM J. Numer. Anal.* **20**, 1 345-357.
- Jea, K.C. and Young, D.M. (1980) "Generalized Conjugate Gradient Acceleration of Nonsymmetric Iterative Methods", *Lin. Alg. Appl.* **34**, 159-194.
- Klein, R.I., Castor, J.I., Greenbaum, A., Taylor, D. and Dykema, P.G. (1989) "A New Scheme for Multi-Dimensional Transfer I. Formulation and 1-D Results", *J. Quant. Spectrosc. Rad. Transfer*, **41**, 199-219.
- Kunasz, P. and Auer, L.H. (1988) "Short Characteristic Integration of Radiative Transfer Problems: Formal Solution in Two-Dimensional Slabs", *J. Quant. Spectrosc. Rad. Transfer*, **39**, 67-68.
- Mesina, M. (1977) "Convergence Acceleration for the Iterative Solution of the Equations $X=A*X+f$ ", *Comp. Meth. Appl. Mech. and Eng.*, **10**, 165-173.
- Ng, K.C. (1974) "Hypernetted Chain Solutions for the Classical Ono-Component Plasma up to $\Gamma = 7000$ " *J. Chem. Phys.*, **61**, 2680-2689 . Olson, G.L., Auer, L.H., and Buchler, J.R. (1986) "A rapidly Convergent Iterative Solution of the non-LTE Line Radiation Transfer Problem" *JQSRT*, **35**, 431-447.
- Saad, Y. and Schultz, M.H. (1985) "Conjugate Gradient-Like Algorithms for Solving Nonsymmetric Linear Systems", *Math. Comp.*, **44**, 417-424.
- Saad, Y. and Schultz, M.H. (1986) "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems", *SIAM J. Sci. Stat. Comp.*, **7**, 856-869.

- Scharmer, G. B. (1981), "Solution to Radiative Transfer Problems Using Approximate Lambda Operators," *Ap. J.* **249**, 720-730.
- Sidi, A., Ford, W.F., and Smith, D.A. (1986) "Acceleration of Convergence of Vector Sequences" , *SIAM J. Numer.. Anal.*, **23**, 178-196.
- Smith, D.A., Ford, W.F., and Sidi, A. (1987) "Extrapolation Methods for Vector Sequences", *SIAM Rev.*, **29**, 199-233.
- Vinsome, P.K.W. (1976) "Orthomin, an Iterative Method for Solving Sparse Sets of Simultaneous Linear Equations", in *Proceeding of the Fourth Symposium on Reservoir Simulation, Society of Petroleum Engineers of AIME*, 140-150.

END

DATE FILMED

01 / 10 / 91