A Matrix Framework for Conjugate Gradient Methods and Some Variants of CG with Less Synchronization Overhead*

Eduardo D'Azevedo† **Victor Eijkhout**‡ **Charles Romine**†

Abstract

We will present a matrix framework for the conjugate gradient methods, which is expressed in terms of whole vector sequences instead of single vectors or initial parts of sequences. Using this framework extremely concise derivations of the conjugate gradient method, the Lanczos algorithm, and methods such as GMRES, QMR, CGS, can be given. This framework is then used to present some equivalent forms of computing the inner products in the conjugate gradient and Lanczos algorithms. Such equivalent formulations can perform all inner product calculations of a single iteration simultaneously, thereby making the method more efficient in a parallel computing context.

1 Matrix Framework

In his 1965 book, Householder [4] presented a short derivation of the conjugate gradient method using a matrix framework. By introducing matrices whose columns are the elements of a vector sequence, e.q. $X = (x_1, \ldots)$, it becomes possible to express statements about sequences as matrix equations. For instance, a Krylov sequence $x_{i+1} = Ax_i$ can be written as $AX = XJ$ where J is the unit lower diagonal matrix $(\delta_{i,j+1})$.

In [3] this matrix framework is used to give derivations of a number of conjugate gradient-like methods, and to derive basic properties of the methods. As an example of the latter, here is a characterization of how Hessenberg matrices arise in iterative methods:

Lemma 1 *If A is a square matrix, R a vector sequence, and $AR = RH$, then H is a non-degenerate upper Hessenberg matrix if and only if the vectors r_i are linear combinations of a Krylov sequence obtained from applying A to a multiple of r_1.*

The proof follows from the fact that taking linear combinations corresponds in the matrix framework to right multiplication by an upper triangular matrix.

* This work was supported in part by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc. and in part by DARPA under contract number DAAL03-91-C-0047

† Mathematical Sciences Section, Oak Ridge National Laboratory, Oak Ridge, TN 37831–6367.

‡ Computer Science Department, University of Tennessee, Knoxville, TN 37996–1301.

In the case of the symmetric conjugate gradient method the Hessenberg matrix generated is triangular:

Lemma 2 *If $AR = RH$ and $r_i^t r_j = 0$ for $i \neq j$, then H is tridiagonal.*

This follows from the fact that $R^t AR$ is both symmetric and upper Hessenberg.

The classic conjugate gradient method follows from factoring the Hessenberg matrix into $H = (I - L)D^{-1}(I - U)$ form, and applying a scaling such that $L = J$. Then the equation $AR = RH$ can be split as $APD = R(I - J)$ and $P(I - U) = R$, where the first equation describes how the residuals r_i are updated by the search directions p_i, and the second equation describes updating the search directions with the residuals.

More sophisticated methods can also be expressed in this framework, for instance the GMRES and QMR methods are based on the following idea. If a sequence R satisfies $AR = RH$, then these methods try to improve on it by setting $g_1 = r_1$ and updating $G(J - E_1) = -ARV$ where J is as above, E_1 is (δ_{i1}), and V is upper triangular. Equivalently, $GJ = R(E_1 - HV)$, so G is minimized by solving a least squares problem for the columns of V. If $H = QU$, $V = U^{-1}\tilde{Q}^t$ where $(\tilde{Q})_{ij} = \begin{cases} q_{1j} \text{if} i \geq j \\ 0 \text{if} j > i \end{cases}$. For the QMR method, which uses the Lanczos method for generating R, H is tridiagonal so U is upper tridiagonal, and $P = RU^{-1}$ can be updated by a simple recurrence. The final iterates are then calculated as $G(J - E_1) = -AP\tilde{Q}^t$, or updating step by step $G(J - I) = -AP\text{diag}(q_{1i})$, which as a vector equation reads

$$g_{i+1} - g_i = -Ap_i q_{1i}.$$

2 Eliminating a synchronization point

The two inner products in the conjugate gradient method present in the case of parallel execution two synchronization points. Since they are interdependent they can not be merged directly. A number of people have tried to overcome this problem. For instance, Saad [7] discovered a recursion relation for the $r_i^t r_i$ inner product requiring $(Ap_i)^t(Ap_i)$ to be computed. Since this can be done in parallel with the $p_i Ap_i$ inner product, by packaging the partial sums only one synchronization point remains.

This method turns out to be unstable, but Meurant [5] suggested a way to stabilize it, at the cost of a third inner product which, however, can be computed simultaneously with the two in Saad's algorithm, so it doesn't introduce an extra synchronization.

Recently the current authors discovered [2, 3] three methods for eliminating the $p_i^t Ap_i$ inner product. These methods rely on computing Ar_i instead of the usual Ap_i, and computing the quantity $p_i^t Ap_i$ recursively.

From $R = P(I - U)$ we find $R^t AR = (I - U)^t P^t AP(I - U)$, or

$$P^t AP = R^t AR + P^t APU + U^t P^T AP - U^t P^t APU,$$

in which the second and third term are strictly upper and lower triangular respectively. Considering the diagonal elements of the left and right hand side we then

find

$$p_i^t A p_i = r_i^t A r_i - u_{i-1i}^2 p_{i-1}^t A p_{i-1}.$$

This enables us to compute the quantity $p_i^t A p_i$ recursively from the actually computed inner product $r_i^t A r_i$.

Additionally, this method and the ones below involve an extra vector update

$$A p_{i+1} = A r_{i+1} + A p_i u_{ii+1}.$$

This is necessary since the rearranged methods no longer compute the matrix vector product Ap_i directly, while it is still necessary to update the residual. Therefore all methods take more scalar work than the traditional algorithm.

Similarly to the above derivation, from the diagonal of $(I-U)^t P^t A P = R^t A R + R^t A P U$ we get

$$p_i^t A p_i = r_i^t A r_i + r_i^t A p_{i-1} u_{i-1i}$$

and from $(I-U)^t P A P = R^t A P = R^t A R (I-U)^{-1}$ we find

$$p_i^t A p_i = r_i^t A r_i + r_i^t A r_{i-1} u_{i-1i}.$$

Since $R^t A R$ is tridiagonal the infinite expansion of $(I-U)^{-1}$ terminates quickly, giving only the indicated recurrence. Both of these methods take an additional inner product ($r_i^t A p_{i-1}$ or $r_i^t A r_{i-1}$), which can be computed simultaneously with the other two.

Stability properties of these methods have been established in [1]: the first rearranged method can be related directly to the analysis in [6], and the second method differs from it by a small amount. However, rearrangement 3 differs by a factor involving a term $r_i^t r_{i+2}$, which is known to be theoretically zero, but in practice potentially non-negligible. Indeed, on certain problems, for instance with extreme gaps in the spectrum of the matrix, this method will fail to converge. Methods 1 and 2 are both stable in practice.

Tests of the rearranged methods with the Harwell-Boeing test set on an Intel iPSC/860 indicate that improvements in performance around 15% are attainable.

References

[1] E.F. D'Azevedo, V.L. Eijkhout, and C.H. Romine. Lapack working note 56: Reducing communication costs in the conjugate gradient algorithm on distributed memory multiprocessor. Technical Report CS-93-185, Computer Science Department, University of Tennessee, Knoxville, 1993.

[2] E.F. D'Azevedo and C.H. Romine. Reducing communcation costs in the conjugate gradient algorithm on distributed memory multiprocessors. Technical Report ORNL/TM-12192, Oak Ridge National Lab, 1992.

[3] Victor Eijkhout. Lapack working note 51: Qualitative properties of the conjugate gradient and lanczos methods in a matrix framework. Technical Report CS 92-170, Computer Science Department, University of Tennessee, 1992.

[4] Alston S. Householder. *The theory of matrices in numerical analysis*. Blaisdell Publishing Company, New York, 1964. republished by Dover Publications, New York, 1975.

[5] Gerard Meurant. Multitasking the conjugate gradient method on the CRAY X-MP/48. *Parallel Computing*, 5:267–280, 1987.

[6] C.C. Paige. Computational variants of the Lanczos method for the eigenproblem. *J. Inst. Maths Applics*, 10:373–381, 1972.

[7] Yousef Saad. Practical use of some krylov subspace methods for solving indefinite and nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 5:203–228, 1984.